



UXI



REVISTA DE SOFTWARE LIBRE DE LA UCI

200

ANIVERSARIO

IV Taller SWL ↑

UCIENCIA

← Humor Libre

/etc/init.d/uxi start ↑



“Publicar nuestro software: único camino hacia la libertad plena”



Jefe Consejo Editorial:



Abel García Vitier
avitier@estudiantes.uci.cu

Editores:



Jorge Luis Betancourt González
jlbetancourt@estudiantes.uci.cu



Lic. Gustavo Javier Blanco Díaz
gustablanca@uci.cu



Félix Daniel Batista Diñeiro
fdbatista@estudiantes.uci.cu



Ing. Rislaidy Pérez Ramos
rpramos@estudiantes.uci.cu

Redacción:



Marisniulkis Lescaille Cos
mlescaille@estudiantes.uci.cu



Dunia Virgen Cruz Góngora
dvruz@estudiantes.uci.cu

Arte y Diseño:



David Padrón Álvarez
dpadron@estudiantes.uci.cu



Ezequiel Manresa Santana
emanresa@estudiantes.uci.cu



Karla Reyes Olivera
kolivera@estudiantes.uci.cu



Yanirys Ortiz Lima
yolima@estudiantes.uci.cu



Elisandra Corrales Estrada
ecestrada@estudiantes.uci.cu

Coordinadores:



Ing. Abel Meneses Abad
abelma@uci.cu

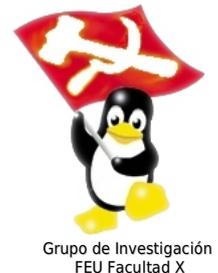


Eiger Mora Moredo
emora@estudiantes.uci.cu



Diana Lilian Antunez Ginarte
dlantunez@estudiantes.uci.cu

Patrocinadores:



Colaboradores:

Revista de Software Libre Proxi
Revista de Software Libre CaS
Proyecto FreeViux
Proyecto NOVA
Proyecto SistClon
Movimiento Código y Letra

Msc. Héctor Rodríguez Figueredo
Aristides Esteban Hernández Guerrero (ARES)
Alex Falco
Yunier Vega Rodríguez
Odaimy Álvarez Acosta
Evelio Clavel Rosales
Marvin Ortega Chacón
Carlos Borroto
Ramón Alexander Anglada Martínez
Mijares Amed Quintero Pajón
Yosbel Brooks Chávez
Yoel Camilo Abreu Ramírez
Anneris Alonso Serrano
Francisco Rolando Díaz Ortiz
Marcos Luis Ortiz Valmaseda



Editorial

Estimado Lector:

Hace 2 años por estas fechas 3 estudiantes de la Comunidad de Software Libre, con el patrocinio del Proyecto Unicornios de la Universidad de las Ciencias Informáticas (UCI), nos dimos a la tarea de crear la primera Revista de Software libre de la universidad, muchos se mostraron excépticos, pero con buen ánimo y ganas de hacer fue publicada el 31 de enero de 2007 la primera edición de lo que sería UXi, nombre que ganó después a partir de propuestas hechas por los miembros de la comunidad. Como fundador de la revista recuerdo lo que comentamos luego de liberar el primer número *"ahora lo más difícil será mantener la revista mensualmente"* y estábamos en lo cierto, desde entonces hasta hoy, ha sido un reto para el equipo editorial obtener artículos relacionados con nuestra temática, nos hemos esforzado mucho para cumplir con la misión de socializar el conocimiento en aras de apoyar el proceso de migración a software libre en la universidad y en Cuba. Con el tiempo UXi fue creciendo en calidad, se incluyeron otras secciones, se amplió el equipo editorial y con cada publicación aumentaba nuestro público lector.

En junio de 2007 la revista es publicada en Internet, a raíz de un artículo sobre la Comunidad de Linuxierra en Buey Arriba, Granma; fue grande nuestra sorpresa al escribir "uxi" en Google y ver los enlaces a sitios comentando sobre la iniciativa cubana. Nos dimos cuenta entonces que en Cuba no existía una revista de software libre (al menos que hubiese publicado más de 3 números) y decidimos dirigirnos hacia un público más amplio. Al comenzar la revista a tener visibilidad desde fuera de la universidad, llegaron las críticas, sugerencias, felicitaciones y suscripciones de todos los lugares de cuba y del mundo. El pasado año UXi celebró su primer aniversario con un festival de software libre, el 2008 constituyó para todos los miembros del staff un año de sacrificios, de cambios en métodos y de concepción de nuevos objetivos. Este año, al cumplirse nuestro segundo aniversario, nos sentimos más motivados y con más ganas de hacer. Lograr que UXi sea considerada Revista Cubana de Software Libre, que gane en seriedad, que sea una revista oficial, que aumenten nuestros lectores y autores, son algunos de nuestros objetivos para el tercer volumen de publicaciones que comenzará en febrero.

No quiero dejar de agradecer a todas aquellas personas que aportaron su grano de arena para que UXi sea lo que hoy es. A todos los que fueron parte de nuestro equipo editorial, a los que aun continúan en esta ardua labor, a nuestros patrocinadores, proyectos amigos de la UCI, de Cuba y del mundo, a los colaboradores que son muchos, a los autores que con total desinterés nos envían sus artículos, y en especial a los lectores para quienes trabajamos y por quienes existimos, a todos ellos les decimos GRACIAS. Estamos seguros que habrá un tercer aniversario y espero para entonces poder hablar de una UCI completamente migrada a software libre, no cejaremos en nuestro empeño de que así sea. Como es costumbre, en nombre del consejo editorial, les deseo una agradable e instructiva lectura.

Recuerde: **"Publicar nuestro software: único camino hacia la libertad plena"**.

Atentamente,
Jefe del Consejo Editorial

Se otorga permiso para copiar, distribuir y/o modificar este documento bajo los términos de la Licencia de Documentación Libre de GNU v1.2 o cualquier otra versión posterior publicada por la FSF; sin Secciones Invariantes, Textos de Cubierta Delantera ni Trasera. Puede

consultar una copia de la licencia en:
<http://www.gnu.org/copyleft/fdl.html>





Contenido



/etc/init.d/uxi start

Un artículo que acoge ideas, sugerencias y proyectos futuros unidos en un solo objetivo: la creación de una empresa cubana de soporte basada en un modelo de negocio que esté regido por la filosofía del Software Libre.

... Página 1



Noticias

... Página 10

Framework para la Gestión de Contenidos Educativos

Interesante propuesta arquitectónica de un framework que garantiza la interoperabilidad entre herramientas para la gestión de Objetos de Aprendizaje.

... Página 13

Aspectos claves para el lanzamiento de Proyectos Open Source

Trabajo para la revisión de los aspectos fundamentales de las comunidades de desarrollo, herramientas que las soportan y aspectos técnicos esenciales a tener en cuenta para el lanzamiento de proyectos de código abierto.

... Página 19



Humor Libre

Para los que no tienen novia, pueden buscar una con Debian...

... Página 25

Propuesta de estudio webométrico sobre impacto y uso del Software Libre en la UCI

Estudio webométrico que permitirá analizar datos exhaustivos obtenidos de los sitios web de la Universidad de las Ciencias Informáticas (UCI), así como el uso del software libre en los mismos.

... Página 26

Entorno de Desarrollo Integrado

En construcción un Entorno de Desarrollo Integrado (IDE) multiplataforma bajo entornos libres para la creación de multimedia.

... Página 30



Eventos

Programa Científico del IV Taller Internacional de Software Libre y Estándares Abiertos de Software

... Página 38



Modelo de negocio basado en la filosofía del Software Libre para una Empresa Cubana de Soporte

Marcos Luis Ortiz Valmaseda
mlortiz@estudiantes.uci.cu
Proyecto Unicornios



Resumen

El presente artículo acoge varias ideas, sugerencias y proyectos futuros unidos en un solo objetivo: la creación de una empresa cubana de soporte basada en un modelo de negocio que esté regido por la filosofía del Software Libre, haciendo énfasis en la utilización de softwares probadamente confiables como son PostgreSQL, OpenBSD y Drupal, los cuales constituirían la base de la misma.

Palabras clave: Modelos de negocio, software libre, empresa de soporte

Abstract

The present article is focused on several ideas, suggests and future projects united with a single objective: the creation of a Support Cuban enterprise based on a business model that follow on the Free Software philosophy making a marked emphasis on the use of highly proved softwares like PostgreSQL, OpenBSD and Drupal, which would be the base of this enterprise.

Keywords: business model, free software, support enterprise

Introducción

Nuestro país está inmerso en una batalla por insertarse en el mercado internacional en materia de desarrollo de software, sector que dominan hasta ahora las grandes transnacionales; así como también en la planificación y materialización de la migración a software libre de la totalidad o la mayor parte de los organismos del Estado, así como los principales sectores del mismo.

Pero si queremos competir realmente con Oracle, Microsoft, Adobe y otras grandes compañías, tenemos que ir a la par o quizás más rápido que ellas –no es un sueño, recuerden que somos muchos los informáticos y técnicos que nos estamos formando ahora, además de los que ya lo son, sólo tenemos que trabajar juntos por un mismo objetivo.

Lo primero que se debe hacer es crear una empresa nacional que brinde soporte al software libre a nivel no sólo nacional sino también a las empresas, ministerios o países del área que requieran de dichos servicios.

Si nos detenemos a pensar un momento sobre este acápite, se comprobará que no es un sinsentido, sino una oportunidad de insertarnos en un mercado que crece aceleradamente en los países del área. Un ejemplo de esto se evidencia en países como Brasil, Venezuela y Ecuador que ya han tomado iniciativas a favor de la migración paulatina a software libre; unos con más resultados que otros, pero ya han dado el paso, lo cual constituye un logro extraordinario.

Desarrollo

Primeramente se debe exponer qué significa que una empresa brinde soporte técnico basado fundamentalmente en las aplicaciones de software libre que se abordarán más adelante.

El soporte técnico es uno de los negocios más exitosos que existen en el mundo informatizado de hoy, y en la práctica ingresan tantas ganancias como las grandes transnacionales productoras de software: Adobe, Kaspersky, Microsoft y SUN Microsystems son solo algunas.

Otras menos reconocidas se han dedicado a darle soporte a las aplicaciones de software libre y han obtenido ganancias; pues este sector de software cada día va ganando más adeptos, y muchas empresas están apostando por él.

¿Qué es el soporte técnico?

El soporte técnico es un compendio de servicios que ofrece una determinada empresa o entidad proporcionando asistencia a un determinado software o hardware. En general, los servicios de soporte técnico tratan de ayudar al usuario a resolver determinados problemas con algún producto; pero dichos servicios no son para capacitar o personalizar, otra de las aristas que se podría explotar.

Existen diferentes tipos de soporte técnico por los cuales se puede ofrecer el servicio: correo electrónico, fax, técnicos, Chat, software de aplicación especializados –un ejemplo de esto son los Help Desk–, pero el más utilizado es el teléfono. El costo del mismo puede variar en dependencia del rango del soporte, ya sea directo, persona a persona, etc. Algunas compañías ofrecen soporte gratuito por medio de foros, salas de Chat, etc.



Esta empresa atendería varias aristas que podrían ser el éxito de la misma:

1. Primeramente tendríamos que enfocar una arista a fomentar el diseño y desarrollo web basados en un excelente manejador de contenido, difundido por todo el mundo: Drupal; ya tocaremos más adelante por qué este CMS (Content Management System) y no otro;
2. Otra arista se centraría en el apoyo al desarrollo de la distribución nacional creada en la Universidad de las Ciencias Informáticas: NOVA Linux, la cual nos ofrecería la enorme ventaja, entre otras, de la cercanía espacial;
3. Otra arista se enfocaría en el desarrollo de una distribución nacional específicamente para servidores basada en OpenBSD, lo cual daría múltiples ventajas sobre otros sistemas operativos en cuestiones de seguridad y criptografía aplicada, temas muy sensibles hoy en día en el mundo informático;
4. Y por último, pero no menos importante: dirigida a la creación de un grupo de desarrollo bien preparado y comprometido en la constante mejora del RDBMS PostgreSQL, y convertirnos en una potencia de desarrollo de este mismo software, para que pueda competir con los mejores sistemas de bases de datos comerciales del mundo, además de ser uno de los grupos de desarrollo más grandes en América Latina, y por tanto brindar soporte especializado al mismo.

¿Por qué hacer énfasis en la utilización de PostgreSQL, OpenBSD y Drupal para que constituyan la base de dicha empresa?

Desglosemos una a una.

¿Por qué PostgreSQL?

No es un mero capricho del autor, sino una necesidad por el crecimiento acelerado y organizado que lleva este paquete de software que todos los días crece y mejora sus excelentes características, las cuales se abordarán en breve; además de que está catalogado como la mejor y más avanzada base de datos Open Source del mundo.



Características de PostgreSQL

Está demostrado que PostgreSQL en sus versiones más recientes puede competir con los grandes productos comerciales como Oracle 10g y Microsoft SQL Server 2008 –la última versión estable es 8.3.5. No por gusto, empresas como IBM, Google, Skype, Deutch Bank y Fujitsu le dan soporte y lo usan en las implementaciones de sus grandes bases de datos; además, también el dominio de Internet .ORG está implementado con dicho software.

El padre de PostgreSQL es el eminente profesor de la Universidad de California (Berkeley para los proyectos públicos) Michael Stonebraker, que lo crea a partir de otro popular proyecto llamado Ingres, el cual estaba basado en la aplicación de las teorías de las bases de datos relacionales. En el año 1986, cambia el nombre del proyecto a Postgres, con el objetivo de aplicar los conceptos de bases de datos Objetos Relacionales.

En 1995, vuelve a cambiar el nombre por Postgres95, continuando el trabajo de Stonebraker dos de sus más prominentes alumnos, que luego derivaría en PostgreSQL, por el hecho de que estos alumnos le incluyeron el **SQL** (Structured Query Language) al mismo. Ya en versiones recientes PostgreSQL cuenta con su propio lenguaje procedural, llamado PL/pgSQL, que comparte algunas características con el PL/SQL de Oracle. En 1996, el proyecto cambia su concepto al mundo del Open Source e inicia su versión 6.0. En el año 2000 se comienza la implementación del soporte para Ipv6. Corre el año 2004, y ya PostgreSQL es reconocido como uno de los mejores motores de bases de datos del mundo, y es la 5ta DBMS más popular en Estados Unidos ese mismo año. En el año 2005, PostgreSQL pasa la prueba de Coverty Inspected, en la cual encontraron sólo 20 errores en 775,000 líneas de código, lo cual constituyó un orgullo y un compromiso para el proyecto.

Ya en el 2006, PostgreSQL llega a su versión 8.0, la cual crece y se mejora por día. Actualmente la última versión estable es la 8.3.3, que mejora muchas características de las versiones anteriores, combinando rapidez con estabilidad, pues a PostgreSQL se le catalogaba de “lento” con respecto a otros productos como MySQL 5.0 y Oracle 10g; el problema es que los que han hecho los benchmark no han probado a PostgreSQL de verdad, con grandes bases de datos de producción, donde realmente se puede ver y catalogar el enorme potencial del mismo. Algunas de las características principales de este excelente producto son:

DBMS Objeto-Relacional

PostgreSQL aproxima los datos a un modelo objeto-relacional, y es capaz de manejar complejas rutinas y reglas. Ejemplos de su avanzada funcionalidad son consultas SQL declarativas, control de concurrencia multi-versión, soporte multi-usuario, transacciones, optimización de consultas, herencia, y arreglos.

MVCC

MVCC, o Control de Concurrencia Multi-Versión (Multi-Version Concurrency Control), es la tecnología que PostgreSQL usa para evitar bloqueos innecesarios. Si alguna vez ha usado algún DBMS con capacidades SQL, tal como MySQL o Access, probablemente habrá notado que hay ocasiones en las que una lectura tiene que esperar para acceder a la información de la base de datos.



La espera es provocada por usuarios que están escribiendo en dicha base. Resumiendo, el lector está bloqueado por los escritores que están actualizando registros.

Mediante el uso de MVCC, PostgreSQL evita este problema por completo. MVCC está considerado mejor que el bloqueo a nivel de fila porque un lector nunca es bloqueado por un escritor. En su lugar, PostgreSQL mantiene una ruta a todas las transacciones realizadas por los usuarios de la base de datos. PostgreSQL es capaz entonces de manejar los registros sin necesidad de que los usuarios tengan que esperar a que los registros estén disponibles.

Write Ahead Logging (WAL)

La característica de PostgreSQL conocida como *Write Ahead Logging* incrementa la dependencia de la base de datos al registro de cambios antes de que estos sean escritos en la base de datos.

Esto garantiza que en el hipotético caso de que la base de datos se caiga, existirá un registro de las transacciones a partir del cual podremos restaurar la base de datos. Esto puede ser enormemente beneficioso en el caso de caída, ya que cualesquiera cambios que no fueron escritos en la base de datos pueden ser recuperados usando el dato que fue previamente registrado. Una vez el sistema ha quedado restaurado, un usuario puede continuar trabajando en el punto que lo dejó.

Altamente Extensible

PostgreSQL soporta operadores, funciones métodos de acceso y tipos de datos definidos por el usuario.

API Flexible

La flexibilidad del API de PostgreSQL ha permitido a los vendedores proporcionar soporte al desarrollo fácilmente para el RDBMS PostgreSQL. Estas interfaces incluyen Object Pascal, Python, Perl, PHP, ODBC, Java/JDBC, Ruby, TCL, C/C++, y Pike.

Lenguajes Procedurales

PostgreSQL tiene soporte para lenguajes procedurales internos, incluyendo un lenguaje nativo denominado PL/pgSQL. Este lenguaje es comparable al lenguaje procedural de Oracle, PL/SQL. Otra ventaja de PostgreSQL es su habilidad para usar Perl, Python, o TCL como lenguaje procedural embebido.

Integridad Referencial

PostgreSQL soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos.

Soporte SQL Comprensivo

PostgreSQL soporta la especificación SQL99 e incluye características avanzadas tales como las uniones (JOIN) SQL92.

Multiplataforma

Corre en casi todos los principales sistemas operativos: Linux, Unix, BSDs, Mac OS, Beos, Windows, etc. (34 plataformas soportadas).

Excelente documentación

Documentación muy bien organizada, pública y libre, con comentarios de los propios usuarios.

Gran comunidad de usuarios y desarrolladores

Comunidades muy activas, varias comunidades en castellano.

Soporte para triggers, procedimientos almacenados, índices, secuencias, etc.

Soporte de todas las características de una base de datos profesional (triggers, store procedures – funciones, secuencias, relaciones, reglas, tipos de datos definidos por usuarios, vistas, vistas materializadas, etc.)

Soporte para SSL

Soporte de protocolo de comunicación encriptado por SSL.

Características adicionales para servicios de alta disponibilidad

Extensiones para alta disponibilidad, nuevos tipos de índices, datos espaciales, minería de datos, etc.

Características para la mejora y optimización de la base de datos

Utilidades para limpieza de la base de datos (**Vacuum**) así como para análisis y optimización de consultas (**Explain**)

Tipo de almacenamiento para datos grandes

Almacenaje especial para tipos de datos grandes (TOAST)

Además de las características de PostgreSQL se deben conocer los límites del mismo para hacer un buen uso del software contando con este aspecto tan importante:

Máximo de base de Datos: ILIMITADO

Máximo de tamaño de Tabla: 32TB

Máximo de tamaño de Registro: 1.6TB

Máximo de tamaño de Campo: 1GB

Máximo de registros por Tabla: ILIMITADO

Máximo de campos por Tabla: 250 a 1600 (depende de los tipos usados)

Máximo de índices por Tabla: ILIMITADO

Número de lenguajes en los que se puede programar funciones: aproximadamente 10 (PL/pgsql, PL/java, pl/perl, pl/python, tcl, PL/php, C, C++, Ruby, etc.)

Métodos de almacenamiento de índices: 4 (B-tree, Rtree, Hash y Gist)



Además de todas las ventajas que se han expuesto aquí, hay que señalar que en Latinoamérica existen muy pocas empresas que le dan soporte a PostgreSQL, y las mismas no dan soporte global, o sea soporte a un ministerio de un país en desarrollo, a grandes compañías con grandes bases de datos en desarrollo, y este sería el punto clave para poder insertarnos en el mercado latino de soporte a PostgreSQL.

En el mundo muchas empresas se dedican a estos menesteres. Algunas de las más reconocidas son IBM (U.S), Fujitsu (Japón), Command Prompt (U.S) y EnterpriseDB (U.S), la cual obtuvo el premio del último Linux World Awards con su producto PostgreSQL Plus, reconocido como la mejor base de datos del 2008, categoría en la cual también estaban nominados: Oracle 10g, MaxDB, JavaDB, etc.

Se puede comprender que somos un rival potencial para estas empresas, debido al número de profesionales de la informática con que contamos en el país, y que podríamos adueñarnos del mercado en el área.

Licencia de PostgreSQL

PostgreSQL está bajo la licencia **BSD** (Berkeley Software Distribution)

PostgreSQL Database Management System (formerly known as Postgres, then as Postgres95)

Portions Copyright © 1996-2005, The PostgreSQL Global Development Group
Portions Copyright © 1994, The Regents of the University of California

Permission to use, copy, modify, and distribute this software and its documentation for any purpose, without fee, and without a written agreement is hereby granted, provided that the above copyright notice and this paragraph and the following two paragraphs appear in all copies.

IN NO EVENT SHALL THE UNIVERSITY OF CALIFORNIA BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, INCLUDING LOST PROFITS, ARISING OUT OF THE USE OF THIS SOFTWARE AND ITS DOCUMENTATION, EVEN IF THE UNIVERSITY OF CALIFORNIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

THE UNIVERSITY OF CALIFORNIA SPECIFICALLY DISCLAIMS ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE SOFTWARE PROVIDED HEREUNDER IS ON AN "AS IS" BASIS, AND THE UNIVERSITY OF CALIFORNIA HAS NO OBLIGATIONS TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

Esta licencia tiene menos restricciones en comparación con otras como la GPL, por lo que su dominio es público. La licencia BSD al contrario que la GPL permite el uso del código fuente en software no libre.

Tengo las 4 libertades del software libre, además:

- Puedo usar Pgsq para fines comerciales de cualquier tipo
- Puedo revenderlo
- Puedo rebautizarlo
- Puedo..... todo

Ahora se tratará lo referente a otro de los aspectos que constituyen la base para la implementación de la empresa: OpenBSD. Primeramente debemos ver qué es OpenBSD. Pueden valorar las ventajas y las desventajas del mismo.

¿Que es OpenBSD?

Es un sistema operativo libre del tipo UNIX, multiplataforma, basado en 4.4 BSD. Es un descendiente de NetBSD, con un foco especial en la seguridad y la criptografía. Este sistema operativo se concentra en la portabilidad, cumplimiento de normas y regulaciones, corrección, seguridad proactiva y criptografía integrada. OpenBSD incluye emulación de binarios para la mayoría de los programas de los sistemas SVR4 (Solaris), FreeBSD, GNU/Linux, BSD/OS, SunOS y HP-UX.

OpenBSD también se distribuye bajo la licencia BSD, aprobada por la OSI. La última versión estable del mismo es la 4.4 del 1ro de noviembre del 2008, siguiendo con su política de liberación cada 6 meses, la próxima estará lista para mayo del 2009.

¿Por qué OpenBSD?

Si existen tantos sistemas operativos que conocemos como Debian GNU/Linux, Red Hat Enterprise Linux, OpenSUSE Linux, FreeBSD, SUSE Linux, CentOS, Ubuntu Server Edition, por sólo citar algunos, que se usan en el país para la administración de servidores, ¿Por qué adentrarnos en uno que no conocemos mucho en nuestro país y fomentarlo?

Muchos administradores de red conocen las potencialidades que tienen los BSDs y los consideran mejor para los servidores que GNU/Linux, por el excelente soporte y adaptabilidad de hardware como una de las tantas razones de lo anteriormente dicho. Se han hecho muchas comparaciones y pruebas en el mundo acerca de si un software tipo servidor funciona mejor en un BSD que en GNU/Linux, y se ha verificado que los BSDs tienen un mejor rendimiento en calidad, por ejemplo, en las operaciones de entrada/salida (I/O) de un base de datos PostgreSQL son un poco más rápidas en OpenBSD 4.1 que en Red Hat Linux Enterprise Linux 5.



OpenBSD es un sistema operativo orientado específicamente a la seguridad, un tema bastante sensible y necesario en la actualidad. Es el sistema más seguro que puede obtenerse libremente sin tener que pasar por registros de exportación que limiten sus características, desde la misma instalación típica del sistema, el mismo está cerrado ante un posible ataque. Sólo el administrador del sistema es capaz de habilitar aquellos servicios que se necesitan.

Las características de seguridad se han cuidado hasta el más mínimo detalle. El sistema soporta por defecto IPSec, varios algoritmos de cifrado (DES, 3DES, Blowfish, Cast, y Skipjack) y funciones hash criptográficas como MD5, y RIPE MD-180.

En lo referente a software hay un problema con OpenBSD: tiene una colección de **ports**, –es decir, de softwares preparados para este sistema– limitada. A pesar de ello, esta colección de **ports** es muy cómoda; y si se dispone de una conexión a Internet basta con montar el CD correspondiente para que automáticamente busque el software, lo baje, aplique los parches necesarios, y finalmente lo compile y se instale en el lugar adecuado.

El hecho de que el volumen de las descargas desde los servidores oficiales de OpenBSD y FreeBSD sea enorme y no contemos con un ancho de banda adecuado constituye, posiblemente, la principal razón por lo que el uso de los BSDs no esté muy difundido en Cuba. Para paliar esta situación podríamos montar un servidor nacional, aunque dicha tarea no constituye, en verdad, nada fácil. Pocos centros en el país usan FreeBSD. Si el programa que se quiere utilizar no está disponible para OpenBSD, se puede obtener un binario para Linux, SCO, o para otro BSD (FreeBSD, NetBSD, o BSDI), y seguramente será posible utilizarlo.

Otra de las ventajas que obtenemos con OpenBSD es el hecho de que es un sistema fácil de aprender en principio; todo está muy organizado, y además, el mismo pasa por auditorías constantes para el código que se incluye en el mismo. Otra sería la licencia sobre la cual es liberado el producto (BSD), lo cual nos favorece en el sentido que podemos construir un producto comercial sin grandes complicaciones legales. Algunos de los productos basados en OpenBSD son:

ePipe by Stallion Technologies

Stallion Technologies usa OpenBSD como sistema operativo integrado en su pasarela de conexión de redes para medianas y pequeñas empresas. Esta pasarela tiene como fin el abaratamiento de los altos costes que supone la conexión de Redes Virtuales Privadas para las pymes, y para proveedores de servicios de aplicaciones que ofrezcan aplicaciones hospedadas a las pymes;

ePipe funciona sobre un procesador Motorola ColdFire 5307 VL-RISC, y usa el software de cifrado integrado en OpenBSD para obtener conexiones seguras en ambas partes de la comunicación.

RTMX

RTMX vende una versión de OpenBSD con complementos de funcionalidades POSIX a tiempo real añadidas. De forma desinteresada, nos ha donado el código fuente para estas extensiones, y pronto serán integradas en OpenBSD.

syscall Network Solutions

Syscall Network Solutions produce el dispositivo de seguridad *syswall*, basado en OpenBSD. Este aparato de seguridad contiene un cortafuegos con un filtro de paquetes con mantenimiento de estado, un sistema de detección de intrusos, y capacidad para Redes Privadas Virtuales (VPN) incluso con direcciones IP dinámicas. Ha sido diseñado especialmente para servicios de seguridad gestionados y servicios de seguridad monitorizados por nuestros expertos en seguridad informática. *Syswall* también se puede usar como un enrutador de ADSL o cable, ya que contiene soporte para PPPoE y DHCP.

También podemos exponer los concisos objetivos del proyecto OpenBSD, por los cuales los desarrolladores se rigen, expuestos por su creador y desarrollador principal Theo de Raadt:

1. Proveer la mejor plataforma de desarrollo posible. Dar acceso a todo el código fuente a desarrolladores y usuarios, incluyendo la capacidad de ver directamente los cambios en el árbol del CVS. Los usuarios pueden incluso mirar nuestro árbol de fuentes y los cambios que se aplican a este directamente desde la web.
2. Integrar el código de calidad de cualquier fuente con un copyright aceptable (preferiblemente del estilo ISC o Berkeley; GPL es aceptable como último recurso, aunque nunca en el núcleo del sistema, y NDA no es aceptable bajo ningún concepto). Queremos que el código fuente se encuentre disponible para que cualquiera lo pueda usar con CUALQUIER PROPÓSITO, y sin restricciones. *Nos esforzamos para que nuestro software sea consistente y seguro, y animamos a las empresas a que usen cualquier parte que quieran.* Existen derivados comerciales de OpenBSD.
3. Prestar atención a los problemas de seguridad y solucionarlos antes de que otros lo hagan (intentar ser el sistema operativo más seguro).
4. Más integración de software criptográfico. Esto quiere decir IPsec, IPv6, máquinas de clave de sesión, Kerberos, free-AFS, y otras formas de criptografía fuerte o de sistemas que hagan uso de esta. OpenBSD se desarrolla y distribuye desde Canadá, y de acuerdo con la ley canadiense es legal la exportación de criptología



a todo el mundo (de acuerdo con el trabajo de un canadiense sobre el tema, y tal y como está documentado en la lista de Control de Exportación de Canadá). Los desarrolladores de OpenBSD participan de modo activo en la investigación y desarrollo de Ipsec.

5. Seguimiento e implementación de normas (ANSI, POSIX, partes de X/Open, etc.).
6. Trabajar para la consecución de un árbol de fuentes muy independiente de la arquitectura. Dar soporte a la mayor cantidad posible de sistemas y hardware.
7. Permanecer lo más al margen que sea posible de políticas de todo tipo. Las soluciones se deben adoptar en base al mérito técnico.
8. No permitir que problemas serios se queden sin resolver.
9. Proveer de una buena plataforma para la compilación/desarrollo cruzado.
10. Importar paquetes externos con modificaciones mínimas, con el fin de que las actualizaciones sean más fáciles. También enviar a los autores los cambios efectuados.
11. Llevar a cabo, aproximadamente cada seis meses, el lanzamiento de una nueva versión en CDROM, sobre todo para recoger fondos para el proyecto...

Plataformas soportadas por OpenBSD

OpenBSD tiene soporte oficial para las siguientes plataformas; generalmente hay versiones preliminares y versiones finales disponibles para estas:

- alpha** Sistemas basados en la arquitectura Alpha de Digital
- hp300** Estaciones de trabajo HP 9000 de las series 300 y 400 de Hewlett-Packard
- hppa** Sistemas Hewlett-Packard Precision Architecture (PA-RISC)
- i386** PC estándar y clónicos basados en la arquitectura i386 de Intel y procesadores compatibles
- mac68k** Apple Macintosh basadas en la arquitectura 680x0 de Motorola, con MMU
- macppc** Máquinas New World de Apple basadas en la arquitectura PowerPC, desde iMac en adelante
- mvme68k** Sistemas VME basados en la arquitectura 680x0 de Motorola
- pegasos** Máquinas Pegasos I de Genesi Sarl. Basadas en PowerPC, placas con chip VIA.
- sparc** Sistema sun4, sun4c y sun4m de la clase SPARC DE Sun.
- sparc64** Sistemas UltraSPARC de Sun
- vax** Sistemas Digitales basados en la arquitectura VAX

OpenBSD es usado por muchas empresas y organizaciones con disímiles fines; desde la construcción de cortafuegos, sistemas de detección de intrusos, hasta implementaciones seguras de Apache, OpenSSH, etc.

Otro uso amplio que se le da a este sistema es como proveedor de servicios de Internet. Podemos poner varios ejemplos de compañías que lo usan con este propósito:

Anonix

Anonix es un proveedor de Internet que ofrece servicios de correo electrónico anónimo y hospedaje de páginas. Todo esto, además de DNS y los cobros, con sistemas OpenBSD.

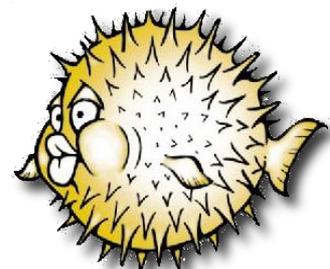
«Confiamos en su seguridad, y en el modo en que está diseñado. La instalación básica no necesita demasiado; estamos seguros de que todo lo que necesitamos para nuestros sistemas se encuentra aquí.»

Anonix Networks Administrators

BS Web Services

BS Web Services, un proveedor alemán, usa servidores OpenBSD para sus DNS primario y secundario (djbdns), hospedaje de web primario (Apache) y servicios de correo primario (qmail-ldap). También usan Autenticación LDAP con OpenBSD (OpenLDAP), además de con bases de datos MySQL. El hostmaster, Henning Brauner, escribe:

«Tuvimos que afinar algo OpenBSD en estas máquinas, en especial maxprocs y maxfiles, pero gestiona cargas enormes con hardware normal. Usamos AMD Athlons (sobre todo los nuevos Thunderbirds) y AMD K6-III. También tenemos algunas máquinas internas con OpenBSD como campos de prueba y servidores de impresión, y para todo tipo de propósitos varios. Vamos a cambiar algunas máquinas más a OpenBSD, en especial nuestros cortafuegos. Desafortunadamente todavía funcionamos con algo de software de código cerrado, pero nos gustaría probar la emulación de Linux. El comportamiento de OpenBSD bajo fuertes cargas, en especial bajo ataques de denegación de servicio, es incomparable con los Linux que usamos anteriormente. Linux funcionaba extremadamente lento, mientras que OpenBSD ni tan siquiera se inmuta (icon el mismo hardware!)».



OpenBSD



Empire Net

Un proveedor de servicios de Internet que usa OpenBSD en sistemas basados en AMD, Intel y Sun, para enrutamiento, cortafuegos, Ipsec (VPN), limitación de bando de ancho, hospedaje de web, servidores de bases de datos, monitorización de redes, detección de intrusos, servidores de correo, servidores para copias de seguridad, servidores de cache, y estaciones de trabajo. Uno de sus enrutadores gestiona el tráfico entre un T3 y ocho puertos fast ethernet, y también entre varias VLANs 802.1Q para separar las redes entre los clientes y las compañías con espacio alquilado. Un servidor de correo OpenBSD gestiona el almacenamiento y el movimiento del correo electrónico, y la autenticación con RADIUS para más de 5.000 usuarios. Varios servidores de web OpenBSD gestionan cada uno de ellos más de 300 sitios web.

Características esenciales de OpenBSD

Vamos a exponer algunas de las características esenciales de este sistema operativo para cualquier entorno de trabajo sencillo:

Protección contra toma de conexiones:

TCP ISN Aleatorio basado en estado de la máquina (ratón, teclado, paquetes recibidos, etc.)

Verificación fuerte del estado de la conexión TCP.

Protección contra ataques de fragmentación.

Entrada al sistema reforzada:

S/KEY integrado de serie en el Login, tanto en /bin/login (consola local, y telnet), como en transferencias de archivos vía FTP.

Opcionalmente se incluye Kerberos IV (KTH-Kerberos) para autenticación con tickets.

A partir de OpenBSD 2.6 no será necesario instalar SSH.

Los servidores desarrollados para OpenBSD han sido comprobados previamente contra posibles problemas de seguridad.

Nivel de Usuario

Generación de números de i-nodo aleatoria (aunque se penaliza ligeramente en la caché de disco).

Se alerta en consola de intentos fallidos o exitosos de realizar un login o un su a la cuenta de *root*.

Se incluye un generador de números aleatorios no predecibles en /dev/urandom.

Distintos niveles de seguridad en las contraseñas. Se puede especificar un algoritmo de cifrado específico para cada usuario y, según el algoritmo, el número de bits que se utilizarán para cifrar la clave.

Solo se pueden obtener datos válidos de `getpwnam(2)` con `uid=0` (*root*) si se utiliza /etc/passwd.db para almacenar las claves de acceso.

Mantenimiento

El proyecto OpenBSD continúa en activo y, a pesar de sus acostumbrados problemas de financiación, lanza una nueva versión del sistema cada seis meses aproximadamente.

Entre dos versiones consecutivas del sistema se suele disponer de 6 a 8 parches de seguridad como resultado del proceso de auditoría continua del sistema. Estos parches se dan a conocer junto con el problema que corrigen a través de foros de discusión públicos como Bugtraq y `comp.security.unix`.

El sistema se puede actualizar en línea mediante ftp, web, ctm, o utilizando cualquiera de los repositorios CVS distribuidos en todo el mundo.

Les dejo entonces que reflexionen sobre lo que se ha expuesto aquí acerca de OpenBSD, y saquen ustedes sus propias conclusiones.

Ahora entramos en el tema de **Drupal**, considerado uno de los mejores CMS del mercado (por no decir el mejor), totalmente libre de licencia bajo GPL, y con amplio soporte para internacionalización en varios idiomas.

¿Qué es Drupal?

Drupal es un sistema de gestión de contenido modular y muy configurable. Es un programa de código abierto, con licencia GNU/GPL, escrito en PHP, desarrollado y mantenido por una activa comunidad de usuarios. Destaca por la calidad de su código y de las páginas generadas, el respeto de los estándares de la web, y un énfasis especial en la usabilidad y consistencia de todo el sistema.

El diseño de Drupal es especialmente idóneo para construir y gestionar comunidades en Internet. Su flexibilidad y adaptabilidad, así como la gran cantidad de módulos adicionales disponibles, lo convierten en adecuado para realizar tipos diferentes de sitio web. El sitio principal de desarrollo y coordinación de Drupal es `drupal.org`, en el que participan activamente varios miles de usuarios de todo el mundo.

Características de Drupal

No voy a poner al detalle las características de Drupal porque son muchas, y sería tema, no para un artículo, sino para varios.

Ayuda on-line: Un robusto sistema de ayuda online y páginas de ayuda para los módulos del 'núcleo', tanto para usuarios como para administradores.

Búsqueda: Todo el contenido en Drupal es totalmente indexado en tiempo real y se puede consultar en cualquier momento.



Código abierto: El código fuente de Drupal está libremente disponible bajo los términos de la licencia GNU/GPL. Al contrario que otros sistemas de 'blogs' o de gestión de contenido propietarios, es posible extender o adaptar Drupal según las necesidades.

Módulos: La comunidad de Drupal ha contribuido con muchos módulos que proporcionan funcionalidades como 'página de categorías', autenticación mediante jabber, mensajes privados, bookmarks, etc.

Personalización: Un robusto entorno de personalización está implementado en el núcleo de Drupal. Tanto el contenido como la presentación pueden ser individualizados de acuerdo a las preferencias definidas por el usuario.

URLs amigables: Drupal usa el mod_rewrite de Apache para crear URLs que son manejables por los usuarios y los motores de búsqueda.

Gestión de los usuarios: Drupal tiene dos características en su contenido que lo hacen uno de los CMS más usados del mercado. La primera es la autenticación de los usuarios; la cual puede ser local o por medio de un servidor LDAP, Jabber, Blogger u otro sitio de Drupal; y la otra característica es la posibilidad de asignar permisos basados en roles; se puede asignar permisos a un "rol" determinado, y agrupar los usuarios por roles, los cuales también se pueden crear fácilmente en el CMS.

Gestión de los contenidos: Esta es otra de las características insignias de Drupal, por las facilidades que tiene el mismo con respecto a este acápite. Ofrece por ejemplo control de versiones, lo cual permite auditar fácilmente y tener una noción exacta del nivel de actualización del sitio; la gestión de plantillas (templates), lo cual permite separar el contenido del sitio de la presentación, cambiar fácilmente el aspecto del sitio y la sindicación del contenido, exportando el contenido en formato RDF/RSS para ser utilizado por otros sitios web.

Administración y análisis: Para la administración y configuración de Drupal no se necesita ningún software adicional, solo un navegador web, además se puede hacer un seguimiento detallado de tipo estadístico viendo por ejemplo la popularidad del contenido, o de cómo los usuarios navegan por el mismo.

Blogging: Drupal es usado como Blogger por sus capacidades; por ejemplo con el soporte de Blogger API permite que un sitio Drupal sea actualizado mediante varias vías como "herramientas web" o de "escritorio". También es conocido que WordPress es considerado mejor Blogger que Drupal. Entonces si se quiere ambos, ya existe un módulo de Drupal para tener Drupal y WordPress como uno solo.

Plataforma: Drupal fue pensado desde un principio para ser multi-plataforma. Funciona lo mismo con un servidor web Apache o con el Microsoft IIS, y en sistemas como BSD, Linux, Windows, MacOSX, Solaris, etc. Posee total independencia de la bases de datos. Actualmente posee soporte para MySQL y PostgreSQL, pero se puede agregar soporte para otro gestor fácilmente.

Otra característica que posee Drupal es la posibilidad de crear un portal multilingüe, con lo cual todo el texto puede ser traducido de forma fácil utilizando una interfaz web, o usando herramientas como GNU ettext.

La Comunidad Drupal: El desarrollo principal del proyecto Drupal está centrado en su sitio oficial <http://drupal.org>. No obstante, existen en el mundo varios grupos de usuarios o comunidades de Drupal (Drupal Hispano es uno de los sitios principales en castellano).

Drupal Japón
Drupal Hungría
Drupal Italia
Drupal Brasil
Drupal Rusia
Drupal Afrikaans
Drupal Alemania
Drupal Francés
Drupal Bélgica
Drupal Polaco
Drupal Rumanía
Drupal Corea
Drupal China
Drupal Taiwán

Rendimiento y escalabilidad: Drupal incorpora un mecanismo de control de congestión que permite habilitar y deshabilitar determinados módulos o bloques dependiendo de la carga del servidor; además del sistema de cache que posee lo cual incrementa el rendimiento y reduce la carga del servidor.

La versiones actuales de Drupal son:

Versión 6.9

Download:

<http://ftp.drupal.org/files/projects/drupal-6.9.tar.gz>

Release notes: <http://drupal.org/node/358987>

Versión 5.15

Download:

<http://ftp.drupal.org/files/projects/drupal-5.10.tar.gz>

Release notes: <http://drupal.org/node/358984>

Rendimiento y escalabilidad: Drupal incorpora un mecanismo de control de congestión que permite habilitar y deshabilitar determinados módulos o bloques dependiendo de la carga del servidor; además del sistema de cache que posee lo cual incrementa el rendimiento y reduce la carga del servidor.



¿Por qué Drupal?

No es mero capricho del autor, sólo que se rige por la tendencia mundial de las grandes compañías al usar este CMS. Dos ejemplos de estas son Yahoo e IBM, las cuales apostaron por Drupal para crear sitios internos y hasta sitios colaborativos.

IBM fue una de las primeras en realizar este tipo de trabajo. Según los expertos de esta compañía:

Louis Weitzman (louis.weitzman@gmail.com)
Senior Software Engineer,
Alister Lewis-Bowen (alister.lewisbowen@gmail.com)
Senior Software Engineer,
Stephen Evanchik (evanchsa@gmail.com)
Software Engineer,

apostan por las tecnologías libres como Eclipse, Apache, PHP, MySQL y Drupal para la creación de sitios medianamente complejos, con un nivel de customización al mínimo detalle.

Ellos explican porqué usan Drupal para la creación de una organización ficticia llamada International Business Council (IBC) para poder explicar el poder que tiene Drupal; la cual tiene como objetivo conectar a los empleados de la compañía con socios externos de negocios en una comunidad de tipo colaborativa.

De ser así, Drupal les da muchas facilidades desde los mismos términos de la Ingeniería de Software, por la excelente arquitectura y organización que presenta el CMS.

Otro elemento que ellos destacan es la posibilidad de agrupar personas en roles o grupos; algo esencial cuando se trata de términos de ingeniería de software y arquitectura de la información. En su ejemplo identifican tres grupos de usuarios: los clientes, líderes de grupo y los administradores; los cuales de una manera natural y fácil se pueden agregar y darle a cada uno las tareas y los permisos correspondientes.

En sentido general, Drupal les ha hecho más fácil llevar el ciclo de vida de un software determinado, en este caso, una aplicación web, por las disímiles características que presenta, y a mencionadas anteriormente, y hacen alusión a las que lo hacen totalmente único en su tipo:

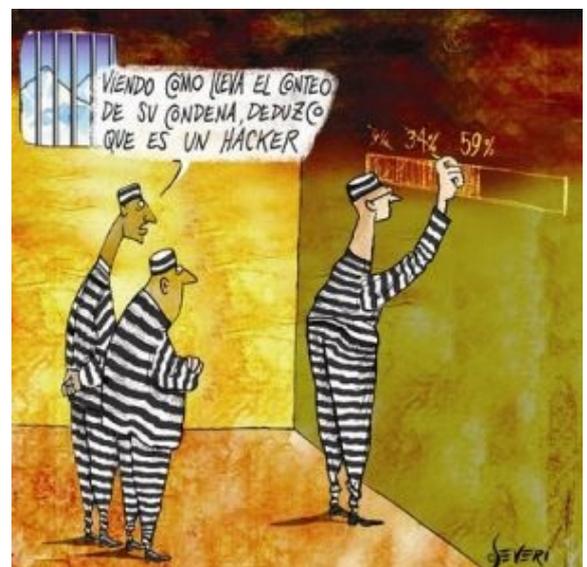
- Control de sesión, incluyendo expiración y firma de términos legales y condiciones
- Búsqueda de contenido
- Separación del contenido de la presentación
- Curva de aprendizaje muy simple del CMS
- Interfaz de administración simple y amigable
- Control de privilegios de acceso
- Enorme comunidad de desarrolladores y usuarios

El último punto a tratar acerca de por qué NOVA Linux como sistema operativo de tipo Desktop será el tema otro artículo.

El presente artículo es una apuesta del autor por el software libre, muchas empresas en el mundo se han decidido por él, recapacitemos cuánto nos falta en este camino.

Bibliografía

- <http://www.postgresql.org>
- <http://www.postgresql.cl>
- <http://www.pgfoundry.org>
- <http://www.openbsd.org>
- <http://www.slony.info>
- <http://www.skype.org>
- <http://www.enterprisedb.com>
- <http://es.wikipedia.org/PostgreSQL>
- <http://es.wikipedia.org/Soporte>
- http://es.wikipedia.org/Help_Desk
- <http://www.daemonnews.org/>
- <http://www.freebsd.org/>
- <http://www.dayrom.com.au/satyricon/>
- <http://www.netbsd.org/>
- <http://www.drupal.org>
- <http://www.ibm.com/developerworks/ibm/library/i->
- <http://www.ibm.com/developerworks/ibm/library/i-osource2>





China castiga con una multa el primer caso de acoso por internet

Un usuario y una empresa informática fueron condenados a pagar una multa por acoso a través de internet, es el primer castigo de este tipo aplicado en el país asiático, según informó el periódico 'China Daily'. El Tribunal Popular del distrito de Chaoyang, en Pekín, condenó a Zhang Leyi y a la empresa 'Beijing Lingyun Interactive Information and Technology' a pagar multas de 5.000 y 3.000 yuanes, respectivamente (731 y 439 dólares, 511 y 307 euros) al demandante, Wang Fei.

Las penas son consecuencia de la denuncia interpuesta por Wang por haber invadido su privacidad y dañado su reputación, después de que la página web mostrara en público información personal del usuario.

Fuente: levante-emv.com

Microsoft extiende la vida de Windows XP, una vez más

Microsoft dijo inicialmente que Windows XP moriría a finales de este año, pero el sistema operativo ha obtenido nuevamente. La nueva fecha límite para la venta de licencias será el 20 de Mayo del 2009.

Esto tiene que ser difícil para Microsoft. Generalmente cuando Microsoft lanza al mercado un nuevo sistema operativo, el antiguo deja de ser protagonista, y es el nuevo el que se pre-instala en las computadoras y las licencias que suelen venderse son las del sistema en "lanzamiento". Evidentemente, los usuarios no suelen actualizar hasta que el nuevo SO de Microsoft lleva al menos un Service Pack, ya que, casi como norma, el nuevo SO suele ser más inestable y con más problemas de compatibilidad que el anterior.

El lanzamiento de Windows 7 está previsto para el 2009, de manera que Microsoft puede que mantenga Xp vivo hasta la llegada de 7, que seguro esperan sea el "salvador" de esta situación.

Fuente: <http://barrapunto.com>

Según el CEO de Mozilla las relaciones con Google se están "complicando"

Mozilla Corp. tiene una larga y "razonable" relación con Google Inc, pero esta se está complicando desde que Google lanzara su propio navegador, según el jefe ejecutivo de Mozilla.

"Tenemos una buena y razonable relación", dijo Jhon Lily, CEO de Mozilla en una entrevista la semana pasada, "pero estaría mintiendo si dijera que las cosas no están más complicadas de lo que solían ser..."



En respuesta a las preguntas acerca del aumento de la competencia del navegador, Lilly también habló de otro rival, Microsoft Corp, por desestimar los intentos de mejorar el rendimiento de un navegador y convertirlo simplemente en una "carrera de resistencia".

Sin embargo, Lilly sigue confiando en que su empresa pueda mantener el impulso que se ha traducido en un incremento del 24 por ciento en cuota de mercado de Firefox desde el comienzo del año.

Google sacó a Chrome de Beta, y un día después de esto cambiaron a Firefox como navegador predeterminado en su Google Pack por el suyo propio, Chrome.

Lilly reconoce que la situación puede parecer extraña, pero defendió el acuerdo con Google. "Estamos cooperando con Google porque creemos que la búsqueda es un punto de entrada fundamental para la Web, y ahora es Google quien ofrece la mejor experiencia de búsquedas".

Fuente: <http://www.historiasdequeso.es/>

Adobe AIR 1.5 llega a Linux

A un mes de su lanzamiento para Windows y Mac, Adobe anunció la disponibilidad de AIR para Linux, compatible con las distribuciones Fedora Core 8, Ubuntu 7,10 y posteriores, además de openSUSE 10,3. Adobe AIR 1.5 es un componente clave de la Plataforma Adobe Flash, que permite que los desarrolladores Web usen HTML, JavaScript, ActionScript y el marco de trabajo de fuente abierta Flex para entregar aplicaciones web fuera del explorador.

La plataforma para desarrollo de aplicaciones de Adobe también cuenta con funcionalidades incluidas en Adobe Flash Player 10, como soporte para filtros y efectos personalizados, transformación y animación 3D nativa y diagramación de texto rico extensible.

Fuente: <http://www.redusers.com/>



BlueSearch recibe el premio Andalucía Sociedad de la Información 2008

La Junta de Andalucía ha galardonado a BlueSearch con el premio Andalucía Sociedad de la Información 2008. BlueSearch un proyecto de Yerbabuena Software, empresa dedicada al desarrollo de software libre. Esto supone un respaldo importante por parte de la Junta de Andalucía para los proyectos de software libre. BlueSearch está disponible en SourceForge y en Morfeo. Anteriormente, otras empresas como la sevillana Emergya (dedicada también al desarrollo en software libre) fueron premiadas. Fuente:<http://barrapunto.com>

Apolo 8: 40 años del primer viaje a la Luna

Apolo 8 fue la primera misión que viajó hasta la Luna y puso a tres seres humanos en la órbita del satélite. Pero esta misión de la NASA, de la que en estos días se cumplen 40 años, es más conocida por una imagen, la fotografía denominada por la NASA como [AS08-14-2383](#), más conocida como 'Earthrise' (Amanecer terrestre), que muestra la salida de la Tierra vista desde su satélite. Aunque aquel 24 de diciembre de 1968 los astronautas de la misión tomaron varias imágenes, 'Earthrise', la primera en color, se convirtió en un hito de la historia de la fotografía.

Fuente:<http://barrapunto.com>

Liberado el kernel 2.6.28 de Linux

Linus Torvalds ha liberado la versión kernel 2.6.28 de Linux coincidiendo con el día de Navidad. Dentro de los cambios más significativas respecto a la versión anterior se destacan: la eliminación de la etiqueta 'experimental' del sistema de archivos Ext4, la integración del gestor de memoria para GPUs GEM (Graphics Execution Manager) que promete toda una revolución en cuanto al soporte gráfico de Linux y que, permitirá, entre otras cosas, el establecimiento de los distintos modos gráficos desde el kernel ('Kernel Mode Setting') así como la unificación de código que, actualmente, se halla disperso y -lo que es peor- duplicado, entre los diferentes drivers de los distintos fabricantes de GPUs, el servidor X y el propio kernel de Linux.

Red Hat obtiene mayores beneficios de lo esperado, pese a la crisis

A pesar de la crisis económica mundial, los ingresos de Red Hat (la empresa de soluciones de código abierto) han crecido un 22% durante el tercer trimestre de 2008 respecto al mismo periodo del año pasado (ascendiendo hasta los 165,3 millones de dólares), superando las previsiones de los analistas, y un 1% respecto al trimestre precedente.

De esta cantidad, 24,3 millones se traducen en beneficio neto. "Nuestros sólidos resultados financieros de este trimestre reflejan nuestra posición fuerte en el mercado como proveedor de soluciones de infraestructuras de alta calidad a bajo precio", ha comentado Jim Whitehurst, presidente y consejero delegado de Red Hat. Red Hat es probablemente una de las empresas que más activamente contribuyen al desarrollo del Software Libre.

2008, un año Open Source

Basado en un artículo publicado en Information Week, se seleccionaron los 10 hitos más importantes relacionados con la comunidad Open Source producidos durante el año 2008:

1. El auge (y precio barato) de los netbooks

Con la etiqueta de "Linux-powered" y un precio muy ajustado, el ASUS Eee PC y el resto de netbooks del mercado no llegan a ofrecer todo lo que un portátil es capaz (principalmente por su limitadas prestaciones en cuanto al procesador, memoria y pantalla utilizado), pero son más que suficientes para todo lo que se les pide.

2. El compromiso de Sun Microsystems



No, Sun todavía no se ha convertido en una nova, pero está empezando a generar mucho calor. A pesar de su caída de ingresos y de que no lo está pasando nada bien con la crisis, su compromiso con el mundo Open Source es una gran noticia.. Está claro que no se podrá decir que no lo están intentado: el patrocinio a OpenOffice, la liberación de Solaris con una versión (OpenSolaris) pensada para trabajar en entornos con escritorios amigables (lo que llaman los americanos "friendly").

3. El lanzamiento de Ubuntu 8.10 y Fedora 10



Aunque sobre "gustos hay colores", tanto Ubuntu como Fedora son dos de las "distros" más importantes del mundo Linux. Ubuntu 8.10 Intrepid Ibex alcanza un nivel de usabilidad y fiabilidad no visto hasta el momento.



4. El lanzamiento de Google Chrome



Cuando Google presentó su navegador Chrome todos nos preguntamos: "¿otro browser más?". Sin embargo, lo cierto es que tras un corto periodo en fase beta (demasiado breve para mucha gente que opina que todavía tiene mucho que pulir), ya está disponible la versión 1.0, con sus desarrolladores presumiendo de cosas como su rendimiento JavaScript.

5. El lanzamiento de Google Android



De nuevo Google, aunque no en solitario, no olvidemos que detrás de Android está la Open Handset Alliance, formada por más de 30 empresas de tecnología y telefonía, demuestra que su compromiso con Open Source no es una simple pose. Android es 100% Open Source con una licencia que permite que cualquiera pueda contribuir con ideas, código o documentación.

6. Nokia activa Symbian



Tras la compra total de Symbian por parte de Nokia, la Symbian Foundation anunció que tiene como objetivo unir Symbian OS, S60, UIQ y MOAP(S) para crear una única plataforma móvil de software abierto.

7. Open Source sí, pero con licencia



Un tribunal de los Estados Unidos estableció que aunque un programa de software pueda ser descargado gratuitamente de Internet junto a su código fuente, no significa que el que lo baje pueda ignorar las condiciones de la licencia adjunta.

8. Hans Reiser es acusado de asesinato en primer grado Hans Reiser era el programador jefe del sistema de ficheros reiserFS y fue el fundador de Namesys, empresa que se encargaba de desarrollar reiserFSv3 y reiserFSv4. Hace varios meses fue acusado de asesinar a su ex-mujer. Después de un juicio que duró casi seis meses y una deliberación de tres días por parte del jurado, finalmente Hans Reiser fue sido declarado culpable de homicidio en primer grado de su ex-mujer Nina Reiser. Como es lógico, todo el mundo se plantea cuál será el futuro de reiserFS v3/4.

9. Debian OpenSSL



Los "guardianes" de la distribución Debian de Linux se llevaron una desagradable sorpresa cuando encontraron que la implementación de su paquete de encriptación OpenSSL, usado para proteger los datos que se transmiten desde y hacia sitios web, tenía un bug. Pero peor que eso fue darse cuenta de que las claves generadas por el OpenSSL de Debian desde 2006 estaban en riesgo. Afortunadamente, el problema fue rápidamente resuelto, ofreciendo instrucciones para explicar cómo generar nuevas claves de encriptación que sustituyeran a las menos fiables, pero el incidente sirvió para que la comunidad Open Source se diera cuenta de que nada está libre al 100% de sufrir problemas de seguridad.

10. SCO pierde con Novell



Después de varios años litigando, una resolución de un tribunal norteamericano obligaba a SCO (declarada en bancarrota en el 2007) a pagar 2,5 millones de dólares a Novell tras el juicio en el que se dirimía la propiedad de Unix. Para el juez, Novell era el propietario real de los derechos de explotación de Unix, un duro palo para SCO, empresa que se había ganado la enemistad de mucha gente por sus "ansias recaudatorias". Esta sentencia también sentó muy bien a IBM, que mantenía una disputa legal con SCO desde que en 2003 ésta la demandará por considerar que había robado el código de Unix.

Framework para la Gestión de Contenidos Educativos

Roxana Cañizares González* , Daymy Tamayo Avila, David Leyva Leyva, Javier Soler Martín
 (*) rcanizares@uci.cu
 Universidad de las Ciencias Informáticas

Resumen

La evolución del uso de las Tecnologías de la Información y las Comunicaciones en la educación ha posibilitado el desarrollo del e-learning y con ello la construcción de herramientas capaces de gestionar contenidos educativos. Uno de los aspectos claves en este ámbito y que requiere de grandes esfuerzos, es la creación de contenidos con calidad. Lo idóneo sería poder reutilizar dichos contenidos y que además estos sean interoperables. Esto sólo es posible si dichos contenidos son estandarizados y diseñados como Objetos de Aprendizaje (OA). Del mismo modo, las herramientas que gestionan OAs deben ser interoperables entre sí. Varios proyectos en los últimos años han comenzado a utilizar estos estándares con el objetivo de lograr sistemas e-learning integrales.

En este trabajo se realiza una propuesta arquitectónica de un framework que garantiza la interoperabilidad entre herramientas para la gestión de OA. La propuesta se basa en una Arquitectura Orientada a Servicios, usando los estándares abiertos de interoperabilidad IMS-DRI, SQI y OKI.

Palabras clave: e-learning, interoperabilidad, objetos de aprendizaje, OKI, servicios web, SOA, SQI.

Abstract

The evolution of the use of information technologies and communications on education has enabled the development of e-learning and thereby building tools capable of managing educational contents. One of the key aspects in this area and that requires great efforts, is the creation of contents with quality. The ideal would be to reuse such contents and also that these ones be interoperable. This is just possible if these contents are standardized and designed as Learning Objects. Similarly, tools that manage LO (Learning Objects) should be interoperable with each other. Several projects in recent years have begun to use these standards with the goal to achieve integral e-learning systems.

This paper makes a proposal of an architectural framework that ensures interoperability between tools for managing Learning Object. The proposal is based on a Service Oriented Architecture, using open standards for interoperability IMS-DRI, SQI and OKI.

Keywords: e-learning, interoperability, learning object, , OKI, , SOA ,SQI, web services.

Introducción

Con el desarrollo de Internet y las Tecnologías de la Información y las Comunicaciones (TIC), han surgido nuevos modelos de enseñanza aprendizaje. Un ejemplo de ello lo constituye el e-learning (electronic learning). El e-learning puede definirse como un conjunto de tecnologías, aplicaciones y servicios orientados a facilitar la enseñanza y el aprendizaje a través de Internet/Intranet, que facilitan el acceso a la información y la comunicación con otros participantes (Red TInet, 2005). Innumerables herramientas de soporte al aprendizaje han surgido con el objetivo de facilitar esta modalidad de enseñanza. Tal es el caso de los Learning Management Systems (LMS) o Sistemas para la Administración del Aprendizaje, los cuales tienen como misión la creación y gestión de cursos y ofrecer seguimiento a las actividades de aprendizaje. Paralelamente, se encuentran los repositorios de objetos de aprendizaje (ROAs), que permiten almacenar y gestionar objetos de aprendizaje (OAs); así como las herramientas de autor que posibilitan la creación de los mismos, utilizando estándares para su descripción y empaquetado.

Los OA tienen su fundamento en la posibilidad de reutilizar contenidos educativos, y aunque en la actualidad no existe una caracterización que logre definirlos con exactitud, sí existe un consenso sobre la necesidad de describirlos y diseñarlos de forma que tanto tecnológica como pedagógicamente puedan ser reutilizados.

De esta forma, puede ser considerado OA cualquier recurso con una intención formativa, compuesto de uno o varios elementos digitales, descrito con metadatos, que pueda ser utilizado y reutilizado dentro de un entorno elearning (López, 2007).

Existen diversos grupos e instituciones que han trabajado en la estandarización del e-learning, y específicamente en estándares que confieran las características necesarias a los OA para que puedan ser reutilizados. Entre estas instituciones se destacan IEEE1, AICC2, IMS3, ADL4, los resultados principales de IMS que están relacionados con los metadatos y el más relevante se puede decir que es el estándar LOM (Learning Object Metadata), que establece cómo deben ser descritos los OA. ADL en coordinación con IEEE, IMS y AICC, diseñaron uno de los estándares más difundidos en la actualidad: el Modelo de Referencia de Objetos de Contenido Compartibles (SCORM) orientado a empaquetar OA descritos mediante LOM.

Los estándares mencionados anteriormente facilitan la interoperabilidad de los OA, así como su búsqueda y recuperación. La interoperabilidad es definida por la IEEE como la habilidad de dos o más sistemas o componentes para intercambiar información y para usar la información que ha sido intercambiada. En el contexto de la interoperabilidad entre aplicaciones e-learning, se destacan los estándares Simple Query Interface (SQI), Open Knowledge Initiative (OKI) e IMS Digital Repositories Interoperability (IMS-DRI), aunque existen numerosas propuestas en la actualidad.

La interoperabilidad en el e-learning constituye una tendencia a nivel mundial, lo cual ha sido posible también gracias al desarrollo de tecnologías informáticas como son los servicios web, y las arquitecturas orientadas a servicio (SOA), que suponen el establecimiento de nuevos mecanismos de comunicación, haciendo posible que sistemas desarrollados en diferentes plataformas y lenguajes de programación, puedan interactuar.

Numerosos proyectos confirman el auge de la integración de sistemas e-learning. Entre ellos se encuentran los proyectos CAMPUS, AGREGA, SUMA, por solo mencionar algunos. La tendencia en este sentido es integrar los sistemas a través de frameworks, la utilización de arquitecturas SOA y de estándares.

Con el objetivo de utilizar las ventajas que proporcionan los LMS, en el año 2005 la Universidad de las Ciencias Informáticas (UCI) comienza a utilizar Moodle. Simultáneamente, son desarrolladas en la Universidad herramientas para gestionar OA, tal es el caso de un repositorio de objetos de aprendizaje (ROA) y una herramienta de autor (ROXS). (Leyva, 2007).

Estas herramientas inicialmente se encontraban aisladas, pero con posterioridad se logró una comunicación utilizando el estándar SQI entre ROA y Moodle (Tamayo, 2007), de forma que es posible realizar búsquedas en el repositorio a partir de Moodle e incorporar paquetes SCORM generados con C2SCORM, otra herramienta desarrollada también en la UCI (Tamayo et al, 2007) (Tito D, B. Hidalgo G, J. Gutiérrez G, Y, 2008). Aunque esta solución permite la comunicación entre estos sistemas, no aprovecha las ventajas de una arquitectura SOA y los estándares de interoperabilidad existentes, dejando además aislada la herramienta de autor ROXS desarrollada.

La propuesta que se realiza está encaminada a resolver la anterior problemática, de forma que queden esclarecidas las bases para el desarrollo de nuevas herramientas, siendo posible incluso su utilización en el desarrollo de cualquier herramienta para el e-learning, y no sólo las capaces de gestionar objetos de aprendizaje, así como su utilización para la integración de otras cuya arquitectura no puede ser cambiada. Para profundizar en esta propuesta puede ser consultada la Tesis de Grado Framework para la gestión de contenidos educativos, de la autora Roxana Cañizares González. (Cañizares, 2008).

En las secciones que siguen se describe la estructura del framework que se propone, detallando los estándares que se utilizan para su concepción. Se proponen algunas pautas para la implementación del mismo y cómo lograr que este alcance diferentes niveles de madurez.

Estructura del Framework

El framework que se propone posee tres capas: presentación, servicios, y acceso a datos, como se muestra en la Figura 1.



Figura 1. Arquitectura por capas del framework

Capa Presentación: es la responsable de interactuar con el usuario de la aplicación. Maneja el contexto del usuario y le permite interactuar con la capa de negocio (servicios) dónde esta implementada toda la lógica de la aplicación.

Capa de Servicios ó negocio: BUS de servicios que publica tanto los servicios necesarios para garantizar la interoperabilidad como otros servicios adicionales. En esta capa se manejará el negocio del sistema. Su responsabilidad es implementar la lógica de negocio que será consumida por la interfaz del sistema y por otros sistemas a través de los servicios compartidos. Dichos servicios se basan en un conjunto de estándares de comunicación, como son XML para la representación de datos, SOAP para el intercambio de datos y el lenguaje WSDL para describir las funcionalidades de un servicio web.

Capa de Acceso a Datos: El acceso a los datos se realiza a través del BUS de servicios. Permite conectar la capa de negocio con los orígenes de los datos, accediendo a los datos que la capa de negocio necesita para funcionar.

En la Figura 2 se detalla la estructura de la capa servicios de interoperabilidad, destacando los diferentes estándares que se utilizan.

Por otro lado, debido a la necesidad de poder integrar también sistemas cuya arquitectura no pueda ser cambiada, como pueden ser las plataformas Moodle o Sakai, se le incorpora al framework una capa gateway o pasarela entre estos sistemas y la capa de de servicios interoperables, con el objetivo de traducir las peticiones que sean realizadas.

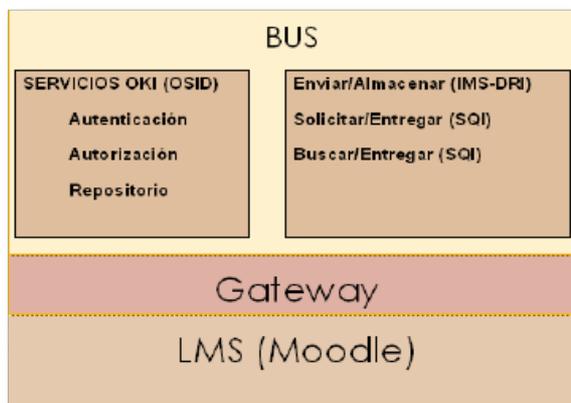


Figura 2. Arquitectura de la capa servicios de interoperabilidad.

Una de las aristas fundamentales en la propuesta, son los estándares de interoperabilidad entre aplicaciones. Se utiliza IMS-DRI para establecer las funciones básicas en un repositorio. Además, para definir los métodos necesarios en el establecimiento de una comunicación entre los sistemas se utiliza SQL y OKI.

A continuación se explican cómo son utilizados cada uno de ellos.

Estándares que conforman el framework

IMS Distributed Repository Interoperability (IMS-DRI).

El estándar IMS-DRI define un conjunto de funciones básicas que se realizan en un repositorio con el objetivo de lograr la interoperabilidad. A su vez, este estándar recomienda cuáles estándares y especificaciones utilizar para implementar dichas funciones. Como bien expresa en su definición, IMS-DRI no obliga a utilizar estas propuestas, por lo que para la implementación de algunas funciones se utilizan otras alternativas que serán explicadas más adelante.

El framework sólo propone utilizar tres de las funciones básicas que define este estándar: enviar/almacenar, solicitar/entregar y buscar/entregar, debido a que alertar/exponer y coleccionar/exponer no son necesarias para establecer una comunicación. Las funciones serán implementadas usando servicios web, los que a su vez formarán parte de una arquitectura SOA.

Para la implementación de enviar/almacenar, se considera la propuesta de IMS-DRI, que identifica mensajes SOAP para el envío de los contenidos. En el caso solicitar/entregar y buscar/entregar, IMS-DRI propone OAI-PMH. Sin embargo, se utiliza el estándar SQL debido a su flexibilidad para el manejo de esquemas de metadatos al abstraerse del esquema utilizado. OAI sólo es posible utilizarlo para repositorios que utilizan DublinCore como esquema de metadatos, principalmente en bibliotecas digitales, y SQL por su parte no obliga a un estándar específico para describir los OA.

A continuación se explican los métodos de SQL que serán utilizados.

Simple Query Interface (SQI).

SQI propone 9 métodos, para establecer la comunicación con otros repositorios, posibilitando así búsquedas federadas de los mismos, y también con otros sistemas externos. A continuación se explica la función principal de estos métodos, que son detallados en la Tesis de Grado Framework para la gestión de contenidos educativos, de la autora Roxana Cañizares González. (Cañizares, 2008).

`SetQueryLenguaje ()`: permite a la fuente tener el control de la sintaxis usada para identificar el lenguaje de la consulta.

`SetMaxQueryResults ()`: define el número máximo de resultados que puede producir una consulta.

`SetResultsFormat ()`: permite a la fuente tener el control del formato de resultado retornado por el destino.

`SetMaxDuration ()`: permite a la fuente establecer un tiempo de consulta en caso de una interfaz de consulta asincrónica.

`ResultsSetSize ()`: define el número máximo de resultados.

`SynchronousQuery ()`: envía una consulta al destino conformado por varios parámetros.

`GetTotalResultsCount ()`: retorna el número total de resultados disponibles por una consulta.

`SetSourceLocation ()`: es requerido antes de que se produzca una consulta asíncrona.

`AsynchronousQuery ()`: permite a la fuente enviar una consulta al destino, mientras los resultados son retornados en un modo asíncronico.

`QueryResultsListener ()`: Este método, iniciado por el destino, devuelve los resultados hacia la fuente.

`CreateSession ()`: crea la sesión, para ello requiere los parámetros `userID` y `password`, retornando un `Session ID`.

`CreateAnonymousSession ()`: crea una sesión sin necesidad de una cuenta en el sistema, devuelve un `ID` de la sesión.

`DestroySession ()`: destruye la sesión, la que se identifica a través del parámetro `sessionID`.

Las guías OSID de OKI.

OKI (Open Knowledge Initiative) es un estándar de interoperabilidad que detalla cómo utilizar los servicios que define y con qué restricciones, es además una propuesta concreta de desarrollo, promueve especificaciones que describen cómo los componentes de un entorno de software se comuniquen entre sí y con otros sistemas. Al definir con claridad los puntos de interoperabilidad, la arquitectura permite que los componentes de entorno de aprendizaje sean desarrollados y actualizados de forma independientes unos de otros. (Thorne & Kahn, 2006).

OKI proporciona un conjunto de interfaces para la definición de servicios denominados OSID's (Open Service Interface Definitions) y proporciona una implementación de referencia para cada uno de ellos, las mismas se caracterizan por una serie de métodos que permiten un proceso en específico, como es el caso de la autenticación. Estas son neutras en cuanto al lenguaje de programación aunque fueron concebidas para Java.

El framework considera tres de las OSID que propone OKI, las cuales son `authentication`, `authorization` y `repository`. A continuación se explica brevemente en qué consiste cada una de ellas. En la Tesis de Grado Framework para la gestión de contenidos educativos, de la autora Roxana Cañizares González (Cañizares, 2008) se detallan los métodos de cada una de ellas.

OSID Authentication:

Soporta invocar procesos de autenticación. Estos servicios son responsables de garantizar cualquier información apropiada para llevar a cabo la autenticación. Los servicios que maneja esta OSID permiten probar si un usuario es autenticado, para ello devuelve el identificador que corresponde a cada usuario, los tipos de autenticación posible y además destruyen todas las autenticaciones o solo aquellas de un tipo dado.

OSID Authorization:

Esta OSID provee una forma de definir quién está haciendo qué y cuándo.

OSID Repository:

El repositorio OSID abarca el almacenamiento y recuperación de contenidos digitales denominado recursos, así como información de los mismos.

Niveles de madurez del Framework.

A continuación se exponen algunos criterios sobre cómo alcanzar hasta el nivel 3 de madurez del framework propuesto a partir de la arquitectura SOA que se utiliza.

Nivel 1 (Punto a Punto). Para lograr un primer nivel, deben implementarse como servicios web las OSIDs authentication, authorization y repository de OKI, los 12 métodos del estándar SQI y la función básica enviar/almacenar de IMS-DRI. Además, es necesario que los consumidores del servicio y quien lo publica acuerden un protocolo mediante el cual se defina cómo es invocado el mismo, cómo se pasan parámetros, cómo se reciben los resultados y cómo se manejan errores, entre otros. Para ello se propone XML y el protocolo SOAP. XML será el lenguaje de etiquetado para describir los datos de los mensajes en formato de documento. SOAP se utilizará para el intercambio de mensajes basados en XML, utilizando http y https. Estos últimos son protocolos para solicitar o responder mensajes entre clientes y servicios, en el caso de https garantizando el envío seguro.

Por otro lado, los servicios deben permitir distribuir los detalles en un lenguaje neutro, proporcionando la información necesaria al cliente para que este pueda interactuar con el servicio, para ello se utiliza WSDL, también basado en XML para describir las interfaces públicas, protocolos y formato de mensajes requeridos para interactuar con un servicio web. Todos estos aspectos son agrupados en un schema, por lo que para consumir un servicio lo primero que debe hacerse es localizar este schema. El documento WSDL está dividido en dos partes, una concreta que define el "cómo" y "dónde" y una abstracta que define qué hace el servicio a través de los mensajes que envía y recibe.

Una vez implementado el servicio utilizando los estándares descritos, su publicación en este nivel se realiza localmente, por lo que al establecer una comunicación con una herramienta específica se debe conocer la dirección exacta de dónde se encuentra publicado cada uno de los servicios. La mayoría de los sistemas que actualmente publican servicios se encuentran en este nivel, lo cual trae consigo algunas desventajas. Por ejemplo, en caso de cambiar la dirección de la aplicación que publica los servicios, las aplicaciones que estén consumiendo dichos servicios deberán ser actualizados, proceso que en ocasiones se hace engorroso, debido a que no siempre se tienen implementados los mecanismos de comunicación que garanticen la actualización continua de cambios.

Esto puede provocar inconformidades entre los clientes que dependan de dichos servicios, además de no tener el control exacto del número de máquinas que se conectan a consumir los servicios.

Nivel 2 (UDDI/ESB).

El segundo nivel tiene como objetivo erradicar las deficiencias del primer nivel, logrando mayor madurez de interoperabilidad. Para ello se recomienda gestionar los servicios a través de ESB, aunque para llegar a este se puede desarrollar en dos etapas. En la primera los servicios se publican en una UDDI, y en la segunda se gestionan a través de un ESB.

Utilizar una UDDI permite descubrir el negocio (o servicio) adecuado, de los miles que pueden estar registrados. Define cómo interactuar con el servicio escogido una vez localizado. Describe los servicios y componentes o métodos de negocio de forma automática (o automatizable) en un entorno seguro, abierto y simple.

Una UDDI es un repositorio donde se encuentran los servicios publicados por diferentes aplicaciones. Esto posibilita que, en caso de cambiar la dirección donde se encuentra un servicio, sólo sea necesario cambiarla en la UDDI. Sin embargo, no se tiene un control de las máquinas que se conectan para consumir un determinado servicio.

Un ESB, por su parte, erradica la desventaja de la UDDI antes planteada, cumpliendo todas las expectativas de lograr una interoperabilidad eficiente y segura.

Nivel 3 (BPM).

El tercer nivel tiene como objetivo principal la administración de los procesos a través de los BPM, ofreciendo así una solución completa, que abarca todo el ciclo de vida de un proceso de negocio: análisis, modelación, ejecución y monitoreo de los procesos.

Pautas para el desarrollo del framework.

Para el desarrollo del framework se propone seguir las siguientes pautas:

- Diseñar herramientas interoperables que gestionen contenidos educativos, de modo que no existan herramientas aisladas.
- Utilizar estándares y protocolos Open Source de amplia aceptación por la comunidad educativa.

- Utilizar una arquitectura SOA, teniendo en cuenta los niveles de madurez propuestos.

- Promover la interoperabilidad entre diferentes aplicaciones, utilizando estándares de interoperabilidad como SQL, IMS-DRI y OKI.

- Promover que las aplicaciones sean creadas desde un principio con dicho framework, para mejorar la integración futura con otras herramientas.

Conclusiones

En un entorno cambiante y diverso como el e-learning, las ventajas de un framework respecto a un producto son evidentes. Un framework permite tener un patrón por el cual guiar todas las herramientas, logrando así una mayor homogeneidad de las aplicaciones y garantizar una mejor interoperabilidad entre las mismas. Con la propuesta arquitectónica presentada, se podrán construir herramientas e-learning completamente interoperables. Finalmente, con el desarrollo de este trabajo se lograron identificar los estándares de interoperabilidad IMS-DRI, SQL y OKI como los más utilizados en el campo del e-learning en la actualidad.

Referencias

Aguirre .S, Salvachúa .J, Quemada J, et all, Mediadores e Interoperabilidad en Elearning, 2007.

Cañizares, R., Framework para la gestión de contenidos educativos, Universidad de las Ciencias Informáticas, 2008. Tesis de Grado. Tutores: MSc. Daymy Tamayo Avila y MSc. David Leyva Leyva.

Duva. E, Massart. D, Simo. M, Simple Query Interface Specification, 2004.

Francesc Santanach Delisau, J. C. D., Pablo Casado Arias, and M. A. Forment Proyecto Campus. Una plataforma de integración, 2007.

Tamayo, A. D, Leyva L, D Tito D, B. Hidalgo G, J. Gutiérrez G, Y. Generación de sitios Web a partir de cursos en la plataforma Moodle. Memorias del VI Simposio Internacional de Tele ? Educación y Formación Continua. Informática 2007.

Leyva L, D; Tamayo A, D; Cid A, A; Coss P, J; Surós V, A y Pernía R, R. Herramientas para la creación y gestión de Objetos de Aprendizaje Reutilizables. UCIENCIA 2006. Universidad de las Ciencias Informáticas. La Habana, 2006



Aspectos claves para el lanzamiento de proyectos de código abierto

Laya del Carmen Rabasa Frómeta

laya@uci.cu

Universidad de las Ciencias Informáticas

RESUMEN

Las comunidades virtuales existen mucho antes del surgimiento de Internet. Estudiosos del tema han creado diversas clasificaciones principalmente en cuanto a si están basadas en espacios físicos o no y según los intereses de sus miembros, a la vez que se subclasifican en cuanto a su orientación en sociales, profesionales y comerciales.

Dentro de las comunidades virtuales existentes en Internet, ocupan un lugar relevante las Comunidades de Desarrollo de Software por cuanto han creado productos y servicios de altas prestaciones. Para la creación de comunidades de desarrollo han de tenerse en cuenta diversos aspectos imprescindibles para intentar comunidades exitosas, tales como: Infraestructura técnica y social, establecimiento de las políticas que regirán el desarrollo, licencias, anuncio del proyecto, entre otros.

Fomentar el desarrollo de comunidades virtuales orientadas a intereses profesionales para el desarrollo de software es uno de los intereses de la estrategia del país en la migración a Software Libre y de la Universidad en particular. En el presente trabajo se hace una revisión de los aspectos fundamentales de las comunidades de desarrollo, herramientas que las soportan y aspectos técnicos esenciales a tener en cuenta para el lanzamiento de proyectos de este tipo. Se pretende sea utilizado como referencia básica para enfrentar proyectos de código abierto en la Universidad e incorporados a la estrategia de soporte a la migración a Software Libre en el país.

Palabras clave: código abierto, comunidades de desarrollo, comunidades virtuales, software libre.

Launching open source project: Key points

ABSTRACT

Virtual communities exist long before Internet emerged, and continue growing. Communities' researchers have been interested in community's classification as two main criteria: place-based communities and members' interest-based communities.

These communities are sub-classified into many other criteria like social, professional and commercial orientation.

As example of professional communities those related with software development are very popular in the Internet since they have created useful products and services. For creating and maintaining an open source project through virtual community, various essential aspects, such as technical, social and political infrastructure, licenses, promotion among others, must be taken into consideration for attempting to achieve successful communities.

To encourage the development of virtual communities for software development is one of the aspects in the country's and university's strategy in migration to free software. This work reviews the fundamental characteristics of virtual communities, as well as the technical aspects to bear in mind for the launch of the open source project. This work is sought to be used as reference to deal with open source project through community effort at the university and as part of the free software migration strategy carried out in our country.

Keywords: development communities, free software, open source project, virtual communities,

INTRODUCCIÓN

Aunque las comunidades virtuales existen mucho antes del surgimiento de Internet no existe un concepto generalizado para las comunidades virtuales, una definición basada en múltiples definiciones las define como: "Ciberespacio soportado por tecnologías de la información, centrado en la comunicación e interacción de sus participantes que genera resultados orientados a los intereses de los participantes y resulta en el fortalecimiento de la interrelación en pos de esos intereses". En la literatura se describen diferentes aproximaciones para clasificar las comunidades virtuales tales como el comportamiento de sus miembros, el propósito de la comunidad o los motivos que incentivaron a sus miembros a pertenecer a ella. Según la clasificación adoptada por algunos autores se establece una primera clasificación de las comunidades virtuales en

cuanto a la orientación social, profesional y comercial, las cuales se van desglosando sucesivamente en temas más específicos. Las comunidades virtuales orientadas a la profesión están dirigidas hacia los profesionales o a la discusión de distintos tópicos relacionados con la profesión. Los profesionales participan en este tipo de comunidades con personas fuera de su entorno laboral u organización que necesitan información similar en su trabajo diario. Se establece a su vez una clasificación en Redes para el aprendizaje y las Redes de expertos.

Hoy en día es común el desarrollo de sistemas de código abierto desarrollados por comunidades de acuerdo a los diversos intereses profesionales. La mayoría de los proyectos de código abierto no resultan exitosos y el éxito de ellos en gran medida depende de la revisión de algunos aspectos técnicos imprescindibles para su lanzamiento.

Sin ánimo de entrar en discusión de términos sobre la historia subyacente entre los términos Software Libre y código abierto, en este trabajo se hablará en términos de código abierto, como una denominación de Software Libre más tentativa a empresas y patrocinadores.

En este trabajo se hace una revisión de los principales aspectos que deben ser revisados para el lanzamiento de proyectos de código abierto, con el deseo de que sirvan como guía para esta forma de desarrollo.

METODOLOGIA

Enfrentar proyectos de código abierto, supone métodos de trabajo distintos en cuanto a compartir información, roles y los diversos aspectos a considerar. Se revisaron comunidades virtuales de desarrollo en Internet, así como bibliografía relacionada con el tema para elaborar una guía de referencia de los aspectos claves a tener en cuenta para la salida a la comunidad de proyectos de código abierto.

Dentro de los aspectos a revisar se determinaron: Aspectos para iniciar el proyecto, Infraestructura tecnológica, Infraestructura social y políticas que regirán el desarrollo, Comunicaciones entre los miembros, Empaquetamiento, actualizaciones y versionado, Aspectos financieros, Manejo de voluntarios en el proyecto y Aspectos legales. A continuación se revisa para cada aspecto los elementos imprescindibles que deben ser tenidos en cuenta.

Aspectos para iniciar el proyecto

Para iniciar el proyecto se recomienda hacer lo que en todo proyecto común se corresponde con el estudio preliminar en XX

cuanto a determinar objetivos y resultados esperados. Se recomienda revisar los siguientes puntos : panorama internacional, preparar el proyecto para la versión pública, escoger un buen nombre, escribir la misión del proyecto, establecer que el proyecto es libre, asegurar la disponibilidad del software, establecer los canales de comunicación, guías para los desarrolladores, documentación, presentación, hospedaje y anuncio del proyecto así como establecer el código de conducta que regirá.

Revisión del panorama internacional

Antes de proponerse un proyecto debe revisarse lo que ya existe. Podemos encontrar proyectos similares en ejecución que reúnan total o parcialmente nuestras ideas. Existen sitios importantes de inscripción de nuevos proyectos que se sugiere sean revisados, por ejemplo SourceForge, Freshmeat, el sitio de la Free Software Foundation, entre otros.

De versión privada a la versión pública

Uno de los puntos más difíciles es transformar la visión de un proyecto de una visión privada a una visión comunitaria. En primer lugar es difícil plasmar en palabras comprensibles a los que llegan al proyecto las ideas, limitaciones, describir las asunciones, estructurar la documentación, organizar el código, etcétera. Para los que llegan, estas acciones son muy necesarias para acortar el tiempo de comprensión y familiarización con el proyecto.

Un buen nombre para el proyecto

Escoger un buen nombre representa la identidad del proyecto. Un buen nombre es aquel que está de acuerdo al objetivo del proyecto y da idea de lo que hace el proyecto, lo identifica únicamente, pronunciable en varios idiomas y preferentemente con dominios .com, .net y .org disponibles.

Escribir la misión del proyecto

Escribir la misión del proyecto ayuda a la claridad del proyecto. La misión es una descripción concreta muy corta del proyecto, se estima que se lee en treinta segundos, y aporta información que se considera importante para atraer rápidamente a otros al proyecto. Escribir la misión deviene en algunos casos ejercicio entre los miembros del proyecto que seleccionan cuáles características y objetivos ponderar dentro de su misión.

Establecer que el proyecto es libre

Es importante para los que llegan leer el proyecto, justo detrás de la misión, qué tipo de proyecto es, si es software libre o de

código abierto. Se recomienda poner claramente el tipo de licencia bajo a cual se va a desarrollar el proyecto. Más adelante se revisa en detalle el aspecto de las licencias.

Estado del desarrollo

Algunos estudios sugieren empezar con algún código funcional. Sin embargo proyectos exitosos han empezado con simples especificaciones de lo que se quiere. Se recomienda al menos tener una lista de las posibles tareas a acometer, de forma tal que los que llegaran puedan apreciar el futuro.

Disponibilidad del software

El software debe estar disponible para ser descargado y estudiado. Debe presentarse en formatos estándares. Muchos proyectos no estandarizan su forma de instalación hasta muy avanzado en el tiempo lo cual hace más difícil para los principiantes la entrada al proyecto.

Canales de comunicación

Desde el inicio se debe concebir los canales de comunicación más convenientes: listas de discusión, foros, canales de chat, RSS así como diferentes tipos de asociación a los diferentes canales de comunicación.

Guías para los desarrolladores

Estas guías devienen más bien sociales que técnicas, explican el cómo interactuar con los desarrolladores, instrucciones para reportar errores, subir actualizaciones e indican las políticas que rigen el desarrollo.

Documentación

La documentación es imprescindible. Es necesaria la existencia de documentos que aporten claridad en la comprensión del proyecto. Una documentación organizada, bien estructurada y sobre todo bien actualizada es un elemento crítico para el buen desarrollo de un proyecto de código abierto.

Esto conlleva esfuerzo considerable. Se recomienda para la documentación inicial : Explicar claramente el nivel de experticidad que se requiere en cada caso, describir claramente cómo instalar/configurar el software, preparar tutoriales, o esbozos de estos para las tareas más frecuentes, identificar las áreas incompletas en la documentación y mantener una sección de preguntas frecuentes las conocidas FAQs (Frequently Asked Questions, por sus siglas en Inglés).

La documentación debe estar disponible tanto online como en los paquetes de la distribución del software. Especial énfasis debe tenerse con la documentación que aportan los desarrolladores para la comprensión del código en sí.

Presentación del proyecto.

El proverbio chino de “una imagen vale más que cien palabras” se aplica en este caso con total acierto. Y es que si el proyecto tiene interfaz gráfica, pantallas de instalación nada mejor que dedicarle un espacio para la promoción del proyecto utilizando este recurso. Esto se considera un elemento de motivación que puede influir en el atractivo hacia el proyecto, de ahí la importancia que debemos prestarle.

Hospedaje

Un aspecto a considerar es el hospedaje del proyecto. Existen sitios que ofrecen hospedaje a proyectos libres como es el caso de Sourceforge . Hospedar un proyecto de este tipo requiere una infraestructura tecnológica para algunos o todos de los siguientes servicios: control de versiones, seguimiento de tareas y errores, listas de correo, canales de chat, RSS , gestor de documentación, etcétera. Cada una de las herramientas de esta infraestructura tecnológica se revisan más adelante.

Anuncio del proyecto

El proyecto debe anunciarse por todos los medios disponibles para ello: Desde otros sitios afines, desde listas de distribución de temáticas similares. Se debe evitar incurrir en problemas éticos que contribuyan a la proliferación de correos no deseados.

La comunicación del proyecto debe escribirse minuciosamente, anunciando la satisfacción de abrir un proyecto a la comunidad, con la mínima información para despertar el interés y proveer la dirección electrónica donde se encuentra disponible.

Código de conducta

En todo proyecto de este tipo, resulta necesario establecer las pautas de comportamiento que regirán la conducta de sus miembros. En el establecimiento y cumplimiento de los códigos de conducta interviene sin dudas el sentimiento de la comunidad . Existen códigos de conducta escritos bajo la licencia Creative Commons Attribution-Share Alike 3.0 como el de la comunidad de Ubuntu que muchos han tomado para sus propios proyectos.

Hasta aquí se han revisado los aspectos mínimos para intentar un desarrollo de este tipo. A continuación se revisan en detalle lo concerniente a Infraestructura tecnológica, Infraestructura social y políticas que rigen el desarrollo, Comunicaciones entre los miembros, Empaquetamiento y actualizaciones, Manejo de voluntarios en el proyecto, Aspectos financieros y Aspectos legales.

Infraestructura tecnológica

En la infraestructura tecnológica para el desarrollo de proyectos de código abierto se combinan un conjunto de tecnologías para la captura e integración de la información. A continuación revisamos lo más relevante de cada herramienta.

Sitio Web

El sitio Web es la vía principal de comunicación del proyecto con el público. Provee información de los objetivos, formas de incorporación al proyecto, acceso a la documentación, etcétera. Al diseño, estructuración y presentación de la información en el sitio Web debe prestársele especial importancia.

Listas de correo

Usualmente la lista de correo es la vía de comunicación con mayor actividad dentro del proyecto. El objetivo de cada lista de correo debe ser revisado con antelación, así como la audiencia a la que va dirigida para evitar en lo posible correos ajenos al propósito de la lista. Se sugiere un buen administrador de listas de correo, así como la definición de las políticas base que rigen la participación en ellas. Ante un crecimiento de la comunidad y número de mensajes generados, debe estudiarse las opciones de hacerlas más funcionales en pos de los objetivos del proyecto.

Control de versiones y seguimiento de tareas

Todo proyecto necesita un control de versiones para obtener el código más actualizado en cada momento, así como apoyar la gestión de las posibles ramificaciones y versionado en el proyecto. Entre las herramientas de control de versiones más comunes están CVS y Subversion . Por su parte las herramientas de seguimiento de tareas permiten seguir el estado de tareas, actualizaciones, incorporación de nuevas características, reportar y dar seguimiento a los errores, entre otras. Dentro de las herramientas más populares en el seguimiento de tareas están Trac y Bugzilla.

La selección de una herramienta u otra en cada caso depende de la preferencia de quien lanza el proyecto.

Chat en tiempo real

Los chats son utilizados para aclarar asuntos en tiempo real. Entre los más utilizados están los canales IRC .

Wikis

Las Wikis son sitios Web que permiten ser actualizados fácilmente por los visitantes. La utilización de las Wikis ha aumentado en proyectos de código abierto desde el año 2000 . Las desventajas principales de las Wiki son la no existencia de principios para la navegación, la duplicación de la información, y que hay poca consistencia en cuanto a la audiencia a la que va dirigida.

Canned Hosting

Aunque no existe una traducción literal para Canned Hosting en español, la traducción más apropiada es la de sitios o plataformas para el desarrollo colaborativo. Reúnen en ellos las herramientas de la infraestructura tecnológica necesarias para montar proyectos de este tipo. Hay dos ventajas fundamentales en usar este tipo de sistemas , la primera es que el servidor tiene capacidad y ancho de banda para soportar los proyectos y la otra es la facilidad para la configuración de los servicios que lo componen. La desventaja fundamental está asociada a que se deben aceptar condiciones del proveedor del sitio aun cuando alguna otra herramienta puede ser mejor para el proyecto. Dentro de las plataformas de desarrollo colaborativo más usadas están SourceForge , Gforge , Savannah , Berlios , entre otras.

Infraestructura social y políticas que rigen el desarrollo

El éxito de un proyecto de código abierto se mide por la capacidad de persistir en el tiempo, sumar nuevos miembros y los resultados obtenidos. En gran medida contribuyen las políticas que rigen la comunidad. Hay diversos estilos en este tema Dictadura benévola, donde las decisiones finales recaen en una persona que se supone use la discusión sabiamente; Democracia por consenso, donde las decisiones son aprobadas por la mayoría y Sistemas de voto cuando el consenso no puede ser alcanzado. En todos los estilos influye la llamada "meritocracia" como la posibilidad de tomar decisiones por los méritos y prestigio obtenidos en el desarrollo del proyecto.

Pensar en las políticas que regirán el desarrollo de un proyecto de código abierto, merece análisis de acuerdo a los objetivos que se tracen los participantes.

Comunicaciones entre los miembros

Aunque son múltiples herramientas que soportan el desarrollo en una comunidad, informar sobre cómo utilizar cada recurso es una actividad fundamental. Se debe establecer la forma de reportar errores, tareas por hacer, anunciar vulnerabilidades. Se sugiere asignar un responsable en la comunidad de revisión de tareas y actividades para evitar, corregir redundancias interpretaciones incorrectas, etcétera.

Los miembros de la comunidad deben prestar especial atención a la forma en que se comunican con el colectivo a través de correos electrónicos o foros de discusión. Se deben evitar mensajes innecesarios, las discusiones que no llevan a ninguna parte, las disputas largas de asuntos irrelevantes, y las disputas privadas. Es importante aprender a manejar las llamadas "personas difíciles" dentro del proyecto. Posiblemente aparecerán personas, que con regularidad, tienen ideas contrapuestas al curso del proyecto, las cuales deben ser manejadas con tacto, cortésmente, exponiendo argumentos con las mejores alternativas para el proyecto.

Empaquetamiento, actualizaciones y versionado

Se debe definir la forma en que se realizarán el empaquetado y el versionado de las distintas actualizaciones. A la hora de liberar una actualización del producto que se desarrolla, se debe revisar cuáles funcionalidades incluir y cuáles son nuevas, cuáles errores quedaron resueltos. Esto forma parte de la documentación para esa actualización. Se debe revisar y establecer el cómo se hará el versionado. Existen diferentes métodos de versionado que deben ser revisados. Debe quedar formalmente escrito cómo se numeran las diferentes versiones del proyecto. Se sugiere la designación de una o dos personas dirigiendo el proceso de sacar una nueva actualización, lo cual se hace de acuerdo a las reglas establecidas en la comunidad para determinar qué se incluye y qué no.

Manejo de voluntarios en el proyecto

Lo más importante en este punto es que cada persona que llega al proyecto sea tratado como un voluntario potencial. Hay múltiples maneras de participar en un proyecto, desde tareas simples de programación, estructuración y actualización de la documentación, hasta desarrollos más complejos. Pero cada persona debe sentir que tiene un espacio para contribuir con el proyecto.

La incorporación de voluntarios requiere especial atención por cuanto aumenta la complejidad del proyecto. Debe revisarse cómo se hace la delegación de tareas, compartir las tareas administrativas lo mismo que las técnicas, quienes aprueban los cambios y cómo hacer las transiciones de responsabilidades cuando un miembro de la comunidad se retira.

Aspectos financieros

Resulta ingenuo pensar que el desarrollo en comunidades está desligado algún apoyo financiero. La comunidad requiere apoyo financiero para costear en muchos casos infraestructura tecnológica, así como acelerar algunos desarrollos. Muchas compañías como es el caso de Sun Microsystems patrocinan directamente comunidades de desarrollo orientando los desarrollos a sus intereses. Muchos de los voluntarios son contratados para trabajar como profesionales en estos proyectos. El pago y contratación a profesionales para desarrollo de sistemas de código abierto, debe estudiarse con sumo cuidado por cuanto puede traer discrepancias entre los miembros de la comunidad. En el aspecto referente a la participación de instituciones en desarrollos comunitarios generalmente se sugiere la presencia descentralizada de sus miembros y que no se vea como un bloque monolítico. Si una organización promueve una idea para el desarrollo debe exponerla tan claramente como sea posible para la comprensión de todos los participantes.

Aspectos legales

Los temas legales en el desarrollo de proyectos de código abierto, involucran asuntos de licencias, derechos de copia y patentes. En el caso de las licencias se debe revisar cuál licencia aplicar al software que se desarrolla. Para esto debe revisarse las licencias libres ya existentes, la compatibilidad con otras licencias libres o propietarias, la protección de la marca, y protecciones de integridad. Dentro de las licencias más utilizadas en este campo son GPL, licencias BSD, MIT, la de Apache, entre otras. Resulta muy cómodo utilizar licencias libres ya establecidas, de esta forma no es necesaria la familiarización con una nueva licencia por parte de los que llegan.

En cualquier caso se recomienda solicitar asistencia legal en el tema. El otro aspecto legal a revisar es respecto al derecho de copia del código generado para la comunidad. Aunque en algunos casos se ignora, se recomienda la elaboración de un acuerdo de licencia con los contribuyentes, CLA, (Contributor License Agreement por sus siglas en inglés). El grado de formalidad y seguridad para la firma de un CLA debe ser consultado con un abogado.

El tema de las patentes de software representa el único aspecto legal contra el cual la comunidad de software libre no puede defenderse a sí misma. Si existe una patente sobre alguna parte del software que se desarrolla la recomendación es simplemente abandonar esa línea de desarrollo. Generalmente, las demandantes son compañías con mejor desarrollo financiero para entablar disputas legales que no pueden ser costeadas por proyectos de código abierto

Se recomienda en cualquier caso la asesoría jurídica para evitar los problemas legales que puedan originarse del uso del producto y/o algunas de sus partes.

Resultados Y Discusión

Para la aplicación de estos aspectos técnicos el equipo inicial que puede ser una o varias personas debe revisar cada aspecto de los aquí enunciados, revisar la experiencia de comunidades similares y tomar las buenas prácticas resultado de la experiencia. Aunque parezca muy intuitivo, la expansión y crecimiento de una comunidad de desarrollo conlleva esfuerzos considerables. Dentro de los aspectos revisados que merecen especial atención están la calidad y disponibilidad de documentación, aspecto que suele ser descuidado en nuestro entorno y que es pieza clave en proyectos de este tipo donde los miembros están generalmente dispersos geográficamente.

El sentimiento de comunidad es un valor intangible donde intervienen las políticas que rigen la comunidad, las formas de comunicación, el cumplimiento de los códigos de conducta donde se pondera el respeto, la camaradería y se rechazan los comportamientos agresivos y soeces. Fomentar el apego al sentimiento de comunidad sin dudas fortalece cualquier proyecto, de ahí la importancia de tenerlo presente en cada momento. Unos cuantos aspectos de los aquí tratados quedan satisfechos mediante un estudio preliminar que logre plasmar en el sitio Web del proyecto, de forma atractiva y organizada, la misión, tipo de licencia, políticas de la comunidad, documentación de todo tipo, así como el acceso a las vías de comunicación y facilidades para familiarizarse con el desarrollo.

El sitio Web es el punto de unión para todos los que se interesen en el proyecto, de ahí la importancia de la estructuración y accesibilidad de la información que se pondrá, el diseño y todo elemento que pueda hacer atractivo el proyecto. Otro aspecto de consideración es el manejo de voluntarios. Lo más deseable para un proyecto abierto a la comunidad es la incorporación de nuevos miembros. Sin embargo, manejar este crecimiento requiere estudio y análisis en la misma comunidad.

La necesaria delegación de tareas, no solo técnicas sino administrativas apoya el crecimiento de la comunidad y su perdurabilidad en el tiempo. Es buena práctica designar determinados roles dentro de la comunidad para la atención de actualizaciones, asignación de tareas, revisión y estructuración de la documentación, entre otros que pueden ser identificados.

CONCLUSIONES

En el artículo se introdujeron características de las comunidades virtuales, haciendo énfasis en las comunidades virtuales de desarrollo de software de código abierto. Se revisaron aspectos a tener en cuenta para el lanzamiento de proyectos de código abierto entre los más importantes se señalan la selección de un buen nombre para el proyecto, la revisión del contexto internacional, esclarecimiento de la misión, la calidad y disponibilidad de la documentación, la selección de una adecuada infraestructura tecnológica, infraestructura social y políticas para el desarrollo, la utilización de las vías de comunicaciones del proyecto así como los aspectos legales principalmente la aplicación de licencias. El lanzamiento de proyectos de este tipo requiere una minuciosa revisión de los aspectos aquí expuestos, haciendo una valoración integral en cada proyecto.

REFERENCIAS

- Apache License. the apache Software Foundation, 2008. [Disponible en <http://www.apache.org/licenses>]
- BerliOS. 2008 [Disponible en <http://www.berlios.de>]
- BLANCHARD A.L., M. M. L. Sense of Virtual Community: Maintaining the Experience of Belonging Proceedings of the 35th Hawaii International Conference on System Sciences, 2002. BSD License. University of California, Berkeley, 2008. [Disponible en <http://www.opensource.org/licenses/bsd-license>]
- Bugzilla 2008. [Disponible en: <http://www.bugzilla.org>]
- Creative Commons Attribution-Share Alike 3.0 license. 2008 [Disponible en: <http://www.creativecommons.org/licenses/by-sa/3>]
- CVS. 2008 [Disponible en: <http://www.nongnu.org/cvs/FOGEL>]
- K. Producing Open Source Software: How to Run a Successful Free Software Project. 2005. 302 p. Freshmeat. 2008. [Disponible en <http://www.freshmeat.net>]
- FSF. 2008a. [Disponible en: <http://directory.fsf.org/FSF>]
- The Free Software Definition, 2008b. [Disponible en: <http://www.gnu.org/philosophy/free-sw.html>]
- Gforge. 2008. [Disponible en: <http://www.gforge.org/projects/gforge/>]
- IRC. 2008. [Disponible en: <http://www.irchelp.org/irchelp/chanlist>]
- Linux Community. 2008. [Disponible en: <http://www.linux.org>]
- MARKUS, U. Charactericing the Virtual Community, 2002. [Disponible en: <http://www.sapdesignguild.org/editions/edition5/cc>]



Mi novia Debian

Empezaremos por buscar una novia
\$ aptitude search novia

Nos la quedamos
\$ aptitude install novia

Ojeamos sus atributos
\$ stat novia

La comparas con otra a la que tanteabas desde hace una semana
\$ cmp novia la_otra

Compruebas si hay conexión
\$ ping novia

Y cómo es dicha conexión
\$ netstat

Sí, definitivamente nos la quedamos. Nos aseguramos que sepa que somos su novio...
\$ chown yo novia

...y de que no nos pueda poner los cuernos...
\$ chmod 700 novia

La moldeamos a nuestro gusto
\$ cat 95-60-90 >> novia

Nos vamos a una fiesta en casa de unos colegas y tenemos un calentón. Nos vamos a un lugar aislado, oculto
\$ cd .dormitorio_padres_colega

Comprobamos que no hay nadie más
\$ ls -a

Buscamos el fastidioso enganche del Sosten...
\$ grep 'enganche_sosten' novia

Al lío. Conectamos nuestros cuerpos de la forma tradicional
\$ ssh yo@novia

Si alguien quiere un 69 (u otro "slot"), tan sólo hay que decírselo
\$ ssh -p 69 yo@novia

Dejamos nuestra semilla
\$ wget http://yo.com/semilla

Y separamos nuestros cuerpos
\$ exit

Ordenamos y limpiamos un poco el cuarto
\$ clear

Poco después te enteras de que metiste el penalti (¡Maldito preservativo!). Nueve meses más tarde tu novia da a luz
\$ tar -xzvf novia.tar.gz

Por las presiones de la familia, decides formalizar la relación y se casan. Fundan una familia con todo lo que ello conlleva...
\$ addgroup familia
\$ adduser novia familia
\$ adduser hijo familia
\$ alias parienta="novia"
\$ alias crío="hijo"

Haces una nota mental para acordarte de este "maravilloso día" y que no se te olvide, todo ello para evitar que no te ponga a parir la parienta
\$ crontab -e

Todo va bien hasta que inexplicablemente un día se le cruzan los cables a la parienta y mata a vuestro hijo
\$ pkill hijo

Meditas sobre todo lo que ha sucedido y sobre las mujeres
\$ man mujeres

Y efectivamente llegas a una sabia conclusión
\$ No existe entrada de manual para mujeres

Pese a que te aseguraste de que no tendrías cuernos, si ambos sois de una raza y el crío de otra, o tienes otro tipo de confirmación, vete buscando a un tal root...
\$ find / -name root

Fuente:
<http://www.tecnologiaslibres.net/2009/01/21/n>



Propuesta de estudio webmétrico sobre el impacto y uso del Software Libre en la UCI.

Liudmila Amat Reyes, Daylenis Hernández Barreiro*
 (*)dbarreiro@estudiantes.uci.cu
 Universidad de las Ciencias Informáticas

Resumen

En el presente trabajo se esboza el diseño teórico de un estudio webmétrico que permitirá analizar datos exhaustivos obtenidos de los sitios web de la Universidad de las Ciencias Informáticas (UCI). Se prevé que sea un estudio diacrónico que muestre la evolución de los elementos estudiados y, en consecuencia, permita ofrecer una visión pormenorizada sobre la difusión y uso del software libre en los sitios de la web de dicha Universidad, así como su desarrollo. Basándonos en esto, podremos observar el estado actual de la migración hacia el Software Libre en la UCI en esta área, establecer pronósticos y tendencias que ayuden a tener un control de las tecnologías que actualmente son utilizadas y organizar líneas de trabajo en función de mejorar el uso de las mismas y acelerar dicho proceso. Este estudio tiene su base en el Polo Productivo Centro de Estudios de Internet (CENTERNET), específicamente en el Grupo de Cibermetría Aplicada (CIBA).

Palabras clave: Estudio webmétrico, migración, software libre, web de la UCI

Abstract

This work is about the theoretical design of a webmétrico study that will allow us to analyze an exhaustive data obtained from several web sites of the University of the Computer Sciences (UCI). We expected it will be a diachronic study that shows the evolution of the studied elements, allowing us to offer a detailed vision about the diffusion and use of free software on the web sites of the current university, as well as its evolution. Basing on it, we will be able to observe the current condition of the migration towards the Free Software in the UCI on this area, after having realized several studies and doing a comparison between them, with the final results to establish predictions and trends that help us to have control of the technologies that are nowadays used, and establish lines of work to improving the use of it. The mentioned study has its base in the Productive Pole, Internet Study Center (CENTERNET); specifically in the Group of Applied Cybermetrics (CIBA).

Keywords: webmetric study, migration, free software, web from UCI

Introducción

En la actualidad Cuba exhibe un desarrollo creciente en la Industria del Software, resultado del impulso al sector de la Informática que se ha planteado la dirección del país; la creación de la Universidad de Ciencias Informáticas es uno de los pasos más sólidos al respecto. En los últimos años nuestro país, en una acción liberadora del monopolio tecnológico y comercial que mantiene el consorcio estadounidense Microsoft, ha apostado por la migración de las tecnologías de la Informática y las Comunicaciones de Sistemas Proprietarios a Sistemas Libres. Esta decisión también se basa, fundamentalmente, en que, aunque los Estados Unidos -país líder en el desarrollo y comercialización de software en el mundo- permitiera la venta de hardware y software a nuestro país, la adquisición de estos sería prácticamente insostenible por parte de nuestra economía, debido a los altos gastos que estos representan por conceptos de registros, patentes y licencias. Esta migración se ha materializado, esencialmente, gracias al Sistema Operativo GNU/Linux. A raíz de lo antes planteado, se hace necesario conocer el impacto y el uso que ha tenido el software libre en su implantación en las diferentes entidades cubanas, como es el caso de la UCI, Universidad pionera en el fomento y desarrollo del software libre en Cuba.

Uno de las vías que podríamos seguir para conocer esta información es el estudio de la web, uno de los principales medios de difusión de la información en estos momentos. En la actualidad las páginas web son contenedoras de programas, archivos comprimidos, documentos de texto y archivos multimedia. Son construidas utilizando diferentes lenguajes de programación y son sustentadas por diversos tipos de servidores web. No estaríamos en un error si afirmáramos que la web es un reflejo de las tendencias que son seguidas por la mayoría de los usuarios en cuanto a la utilización de las tecnologías de la Informática. Observando y estudiando la web, podríamos obtener una visión sobre cual ha sido el uso de los formatos libres frente a los propietarios.

Teniendo en cuenta lo antes expuesto, y basándonos en la Webmetría, disciplina métrica muy joven en su explotación que se encarga del estudio de los aspectos cuantitativos de la construcción y uso de los recursos de información, estructuras y tecnologías en la web, desde perspectivas bibliométricas e infométricas, nuestro trabajo tiene como finalidad el diseño de un estudio webmétrico diacrónico, con el objetivo principal de obtener una serie de indicadores que permitan establecer las tendencias que son seguidas por la mayoría de los usuarios en cuanto a la utilización de los formatos libres sobre los propietarios, dando la posibilidad de conocer la difusión y el uso del software libre y su evolución en esta área.

Al obtener esta información, se dispondrá de elementos que pueden servir como base para emitir un aproximado del estado de la migración que se lleva a cabo en la Universidad, aunque es muy importante destacar que no bastará un solo estudio realizado para obtener suficientes elementos que nos indiquen el comportamiento de esta migración. Es necesario contar con otros estudios que se realicen en un período de tiempo determinado para poder comparar los datos que se obtengan, y entonces así, conociendo la evolución de los elementos estudiados, podremos conocer el estado de dicha migración y será posible trazar estrategias que contribuyan a acelerar este proceso. En una primera etapa el estudio tendrá como punto de partida la web de la UCI, y las pretensiones son de extenderlo a todo lo largo y ancho de nuestro país. Este estudio tiene su base en el Polo Productivo Centro de Estudios de Internet (CENTERNET); específicamente en el Grupo de Cibermetría Aplicada (CIBA).

Metodología

Para alcanzar los objetivos que persigue el estudio se creó una metodología capaz de alcanzar los indicadores que facilitarán analizar los datos acerca del uso del software libre en la web de la Universidad. Esta metodología se estructura en 4 fases que se describen a continuación.

Definición de los indicadores webmétricos.

En esta fase se definieron los indicadores webmétricos que serán contemplados en el estudio. Gran parte de ellos tuvieron su base en una investigación similar a esta llevada a cabo por el Grupo de Investigación en Recuperación Avanzada de Información (Grupo REINA).

Como resultado final se definieron 10 indicadores, agrupados en 4 categorías. Estos indicadores se describen detalladamente en la sección siguiente.

Recogida de datos.

Para la recogida de datos nos apoyaremos en el Grupo de Recuperación de Información, perteneciente al polo CENTERNET, que ha desarrollado un Spider, robot que igualmente permite analizar los documentos hipertexto recorriendo toda la red a través de los hiperenlaces que los conectan. En este caso particular, las informaciones de interés para nuestro grupo recuperadas por el robot son, específicamente, las respuestas de los servidores a las peticiones HEAD y las URLs de los elementos que sean indexados y recuperados por el robot.

Evaluación y representación de los datos.

La información se presentará de dos maneras: una orientada a la descripción objetiva mediante gráficos, los que ilustrarán los resultados de cada indicador, y otra orientada a la valoración mediante textos, donde se realizará la interpretación de los datos para cada uno de los indicadores, lo que posibilitará identificar las tendencias que son seguidas por la mayoría de los usuarios en la web.

Impacto y uso del Software libre en la web.

En esta fase, tomando en cuenta los análisis cualitativo y cuantitativo de los datos obtenidos, se ofrecerá una visión acerca del uso y el impacto del software libre en la web de la Universidad.

Indicadores webmétricos.

El estudio se basará, fundamentalmente, en varios indicadores webmétricos que tienen como fin mostrar estadísticas a partir de las cuales sea posible realizar un análisis cualitativo de las mismas. Los indicadores seleccionados se recogen en cuatro temáticas fundamentales: Sistemas Operativos, Servidores Web, Programación Web, y Formatos de Ficheros.

Los indicadores que se recogen en cada temática tienen un objetivo específico; a continuación expondremos cada indicador con sus objetivos, recogidos por la temática que los agrupa.

1. Sistemas Operativos.

Una de las informaciones más interesantes a considerar en la exploración de la web de la Universidad es la referente a los sistemas operativos sobre los cuales la misma está sustentada. Esto nos dará en gran medida el uso de cada sistema operativo que la conforma. Los indicadores que se recogen en esta temática son:

1.1. Sistemas operativos por número de hosts.

Conociendo el número de hosts que utiliza cada variedad de sistema operativo, dejando de lado aquellos de los que no tenemos datos precisos como puede ser el caso de máquinas que no ofrezcan datos relacionados con este indicador o que ofrezcan datos demasiado genéricos, como podría ser el caso de los que señalaran como Sistema Operativo UNIX, debido a que hay determinadas variedades de UNIX que son consideradas propietarias, podremos apreciar la relación de uso entre los sistemas operativos libres y los propietarios, o sea, ver cuál sistema es más utilizado en la web.

1.2. Sistemas operativos por número de páginas.

Solo no sería interesante conocer la cantidad de host por cada sistema operativo presente en la web, sino la cantidad de páginas albergadas por cada uno de ellos. Esto se debe a que en ocasiones aunque un determinado sistema operativo sea el dominante en cuanto a factibilidad en la web, no significa que sea el más utilizado o preferido en relación a la cantidad de información a publicar. Hay casos en que un determinado sistema operativo es el más presente en la web, pero para la publicación de un volumen demasiado grande de información el más utilizado es otro.

1.3. Distribuciones GNU/Linux por número de hosts.

En el caso de los sistemas operativos libres, sería de vital importancia conocer qué variantes o distribuciones son las más utilizadas en los servidores web de la Universidad. Este indicador nos proporcionaría una base para establecer las tendencias que se siguen en la actualidad en cuanto al uso de las distribuciones GNU/Linux.

2. Servidores Web.

Otra de las informaciones útiles que podemos obtener es la referente al servidor web que utilizan los diferentes hosts de la Universidad. Teniendo como base los indicadores que se analizan en esta temática, podremos observar tendencias generales, así como el mayor o menor grado de implantación de software libre frente a propietario. Tendremos en cuenta para valorar lo antes mencionado los siguientes indicadores:

2.1. Servidor web por número de hosts.

Mediante este indicador pretendemos recoger el número de hosts que utiliza cada tipo de servidor presente en la web de la Universidad, esto nos posibilitará en gran medida conocer qué tipo de servidor es más utilizado en la Universidad.

2.2. Servidor web por número de páginas albergadas.

Observando el tipo de servidor web desde el punto de vista de la cantidad de páginas albergadas, en lugar del número de hosts, podríamos saber con bastante veracidad sobre qué tipo de servidor se concentra el grueso de la información publicada en la web de la UCI.

2.3. Servidor web por Sistemas Operativos.

Mediante este indicador podremos obtener una relación de los tipos de servidores presentes en la web y los sistemas operativos sobre los cuales están montados.

2.4. Servidor web por tamaño de servidores.

Otro asunto es el tamaño de los servidores, el cual mediremos con los datos que dispondremos en ese entonces: el número de páginas. Aunque ya hemos comentado sobre una visión global que nos mostraría el tipo de servidor que más domine en la web, quizá una visión algo más pormenorizada podría hacernos ver algo más allá. Para ello consideraremos varios grupos de servidores en función de su tamaño; a partir de ahí, podremos observar para qué volumen específico de páginas se usa cada tipo de servidor.

3. Programación Web.

Una de las formas de aportar interactividad a las páginas web es el uso de scripts que corren en el servidor, y que reciben datos a través de formularios y similares. Nos hemos planteado como meta en esta temática recoger datos sobre los lenguajes de programación web más utilizados en la Universidad.

Ello nos permitiría conocer en gran medida las tendencias sobre el uso de soluciones de código libre en este campo.

Para recoger los datos antes mencionados haremos uso del indicador: Lenguajes de programación utilizados en el servidor.

4. Formatos de Ficheros.

Las páginas web enlazan recursos de muy diversos tipos, y los formatos de dichos recursos son un exponente claro de la penetración o no de los formatos abiertos.

En su mayor parte, se trata de recursos que se desean difundir de forma general, como pueden ser resultados de investigaciones, material docente de las diversas asignaturas que se impacten en la Universidad hasta archivos comprimidos, archivos multimedia como videos, música, imágenes etc. Cuando no es así, se utilizan protecciones con contraseñas u otros métodos y, en consecuencia, caen fuera del ámbito analizado por nosotros. Naturalmente, tales recursos pueden haber sido elaborados con programas diversos, pero lo importante para este trabajo son los formatos con que se difunden para conocer la penetración de los formatos libres. Estos tipos de elementos enlazados en la web se conocerán mediante el indicador:

Tipos de formatos enlazados.

El estudio será materializado utilizando algunas de las tecnologías y herramientas que son empleadas actualmente por el grupo CIBA, a continuación se mencionan las mismas:

- Sistema Operativo: *Debian GNU/Linux 4.0 (Etch).*
- Sistema Administrador de Bases de Datos: *PostgreSQL 8.1. PgAdmin III (gestor).*
- Entornos de Desarrollo Integrado: *Eclipse.*
- Lenguaje de Programación: *Perl 5.8.8.*
- Módulos de Desarrollo: *Varios Módulos Perl de la CPAN.*
- Gestión del estudio: *DotProject 2.0, Subversion 1.4, RapidSVN 0.9 (cliente).*

Conclusiones

A lo largo del trabajo se ha esbozado un estudio webmétrico que tiene la finalidad de mostrar el impacto y uso del software libre en la UCI. Se articuló una metodología capaz de guiar la realización de dicho estudio y de obtener los objetivos propuestos, así como también se definieron los indicadores que medirán los elementos que serán estudiados y las tecnologías y herramientas que serán usadas para concretar la realización del estudio. Aunque dicho estudio fue concebido para realizarse en la web de nuestra Universidad, proponemos extenderlo a las demás provincias del país para poder conocer cuál es el impacto y uso del software libre en la web en nuestro país.

Referencias

Cristina Fobas Pérez, Vicente P. Guerrero Bote, Félix de Moya Anegón. Fundamentos y Técnicas Cibernéticas.

Artes Gráficas REJAS (Mérida), 196.

Carlos G. Figuerola, José Luis Alonso Berrocal, Ángel F. Zazo y Emilio Rodríguez. El Uso de Software Libre en los Sitios Web Universitarios Españoles.

http://reina.usal.es/index.php?option=com_content&task=view&id=59&Itemid=62



Entorno de Desarrollo Integrado

Yonnys Pablo Martin Olivera

ypmartin@uci.cu

Universidad de las Ciencias Informáticas

Resumen

En la actualidad las nuevas tecnologías de la información y las comunicaciones se han extendido a todos los rincones del mundo, y están causando un impacto revolucionario en los medios de enseñanza tradicionales. Con estos nuevos modelos educativos se va haciendo un énfasis mayor en las universidades y aulas virtuales, donde los estudiantes aprenden e interactúan con software educativos relacionados con las materias que reciben. En Cuba existe una creciente necesidad de productos educativos, entre ellos multimedia. Dichos productos se desarrollan muchas veces usando herramientas de autor propietarias sin licencias, lo que impide su comercialización y exportación. Es significativo destacar que la mayoría de estas herramientas solo permiten crear contenido para una sola plataforma, esto rompe con la estrategia de migración a software libre que se ha trazado el país en vista de lograr la independencia tecnológica. Actualmente se encuentra en construcción un Entorno de Desarrollo Integrado (IDE) multiplataforma bajo entornos libres para la creación de multimedia. Este IDE ha sido dividido en tres subsistemas encargados de definir todas las características de este tipo de herramientas. Subsistema de gestión visual para el diseño de interfaces, Subsistema de gestión de código para edición de código multimedia y subsistema de clases, componentes y ejercicios reutilizables para la creación de multimedia.

Palabras clave: IDE, Herramientas Libres, Multimedia, Software Libre.

Abstract

Nowadays, the new information and communications technologies have spread to all corners of the world, and they're causing a revolutionary impact on traditional ways of teaching. These new educational models are becoming quite useful on universities and virtual classrooms, where students learn and interact with software-related educational materials they receive. In Cuba there is a growing need for educational products, including multimedia.

Such products are often using copyrighted tools without licenses, thereby preventing their marketing and export. It is significant to emphasize that most of these tools only allow you to create content for a single platform, this breaks with the strategy of migration to free software chosen by our country to achieve a true technological independence. An Integrated Development Environment (IDE) for creating multi-platform multimedia environments is currently under construction. This IDE has been divided into three sub-systems: Management subsystem for visual interface design, management subsystem code for code editing and multimedia subsystem classes, exercises and reusable components for creating multimedia.

Keywords: free software, free tools, Integrated Development Environment, multimedia.

Introducción

El mundo avanza cada día a una velocidad mayor, las nuevas tecnologías quedan obsoletas en breves períodos de tiempo. En la actualidad la información y las comunicaciones se han extendido a todos los rincones del mundo, causando un impacto revolucionario en todas las esferas sociales, políticas, económicas y particularmente sobre los medios de enseñanza tradicionales. Las computadoras y sus métodos cada día más interactivos y variados de aprendizaje, la posibilidad del acceso al conocimiento global sin restricciones de tiempo, ni espacio; la dependencia tecnológica de la sociedad y las facilidades que esta brinda, así como todas las ventajas asociadas hacen que cada día el desarrollo de estos tipos de productos dirigidos a este sector en particular sean cada vez más cotizados y necesarios.

Hace ya algún tiempo nuestro país se dio a la tarea de revolucionar y mejorar el sistema de aprendizaje, para lograr esto se han producido una serie de programas que han ayudado en la preparación y la motivación de estos, así como el desarrollo de diferentes programas que contribuyan a esta preparación.

El embargo económico y político del gobierno de los Estados Unidos de América hacia nuestro país también ha afectado seriamente en el avance del desarrollo de soluciones informáticas para la educación. La mayor parte de las grandes compañías productoras de herramientas y materiales para este sector son de origen norteamericano; y las trabas que nos impone el gobierno norteamericano para el uso libremente de estos productos nos lleva experimentar la necesidad de incentivar la producción nacional a esta rama de la producción de software basándonos en nuevas soluciones. Pudiendo así solventar nuestras necesidades, y porque no, también abrimos un poco más a la comercialización o instauración de nuestros productos en el exterior.

En toda Cuba se lleva a cabo un plan de migración a software libre, y específicamente en la Universidad de las Ciencias Informáticas (UCI). Este proceso de migración repercute negativamente en la producción de software educativo en formatos multimedia debido a que en ese momento las herramientas con las que se cuenta son propietarias. Para lograr establecer satisfactoriamente una migración gradual al software libre en la UCI sin que esto repercuta negativamente en la producción de software educativo en formato multimedia la dirección UCI en conjunto con la dirección de Software Educativo de la misma se han planteado la necesidad de elaborar o construir una herramienta que posibilite crear software educativo con tecnología multimedia en menor tiempo, con gran calidad y lo más importante con perspectiva libre, es decir para múltiples plataformas.

Desarrollo

La UCI, al ser una universidad relativamente joven, no cuenta con experiencia en el desarrollo de una herramienta para el desarrollo de software educativo multimedia, por lo que se realizó un estudio profundo del estado del arte de las herramientas libres existentes para el desarrollo de este tipo de productos, como resultado se encontraron varios proyectos que han intentado desarrollar una aplicación alternativa a Macromedia Flash, ahora Adobe Flash, sin embargo, algunos nunca han sido completamente implementados, y otros quedaron solo en la etapa de proyección. Esta investigación demostró la necesidad de desarrollar un software que integrara las actuales herramientas disponibles para desarrollar software multimedia, ajustándolo a las necesidades de la UCI y del país en general. Para facilitar el proceso de desarrollo, debido al nivel de complejidad de la aplicación que se desea implementar, se dividió el sistema en tres subsistemas:

- Subsistema de Gestión Visual.
- Subsistema de Gestión de Código.
- Subsistema Ejercicios y Clases Reutilizables.

Una vez desarrollado cada uno de estos módulos, se integrarían en uno solo, brindando una completa solución para el desarrollo de software educativo con tecnología multimedia en la Universidad de las Ciencias Informáticas. En los siguientes epígrafes se describe cada subsistema por separado y se detallan sus características. El sistema desarrollado tiene como rasgos fundamentales:

- Multiplataforma, es posible compilarlo y obtener ejecutables para la mayoría de las plataformas existentes (Mac OSX, Microsoft Windows, GNU/Linux).
- Basado en plugins, para extender su funcionalidad sin tener que modificar el código base del sistema.
- Cuenta con un conjunto de clases y componentes reutilizables para agilizar el desarrollo de productos multimedia, escritos en ActionScript 2.0 (Crucigramas, Acrósticos, Sopa de Letras).
- Posee un editor visual para la construcción de la interfaz y un editor de código para escribir código específico de multimedia.
- Es software libre, por lo que es posible adaptarlo a las necesidades de desarrollo de cualquier institución.

Herramientas y tecnologías utilizadas:

Para la construcción del sistema se utilizaron un conjunto de herramientas y tecnologías, siempre cumpliendo con el requisito que todo lo utilizado debía ser software libre, y así no crear un sistema comprometido con licencias comerciales que dificultaran su posterior distribución o utilización en nuestro país, a continuación se describe cada una de ellas.

Librería gráfica para el desarrollo: Qt4

Qt es un producto de la empresa noruega de software Trolltech, la cual se dedica a desarrollar librerías y herramientas de desarrollo de software, además es experta en servicios de consultoría. Qt es un conjunto de librerías para el desarrollo de aplicaciones GUI, escritas en código C++, y son completamente orientadas a objetos.

Estas librerías comenzaron a distribuirse comercialmente en 1996 y desde entonces han sido la base para numerosas aplicaciones, incluyendo la popular interfaz gráfica para Linux llamada KDE, disponible en todas las grandes distribuciones de Linux (Jasmin Planchette & Mark Summerfield 2006). Actualmente ha evolucionado hasta convertirse en un completo framework para el desarrollo de aplicaciones, contando con módulos para el manejo de XML, base de datos, gráficos 2D, OpenGL, programación multihilos, etc.

En el mercado se pueden encontrar las siguientes distribuciones de Qt:

- Qt Enterprise Edition y Qt Professional Edition, disponibles para el desarrollo de software con fines comerciales. Incluye servicio de soporte técnico y están disponibles ampliaciones.

- Qt Free Edition, es la versión para Unix/X11 para el desarrollo de software gratuito y de código abierto. Se puede obtener gratis sujeto a los términos de la Q Public License and the GNU General Public License. Para plataformas Windows también está disponible la versión Qt non comercial.

Para el desarrollo de nuestra solución se utilizó Qt Free Edition, que como se indica arriba, permite la creación de software libre.

Características:

- QT es una librería para la creación de interfaces gráficos. Se distribuye bajo una licencia libre GPL(o QPL) que permite incorporar QT4 en aplicaciones open-source.

- Se encuentra disponible para una gran número de plataformas: Linux, MacOS X, Solaris, HP-UX, UNIX noc X11. Además, existe también una versión para sistemas empostrados.

- Es orientado a objetos, lo que facilita el desarrollo de software. El lenguaje para el que se encuentra disponible es C++ aunque han aparecido bindings a otros lenguajes como Python.

- Es una librería que se basa en los conceptos de widgets (objetos), Señales-Slots y Eventos (ej: clic del ratón).

- Las señales y los slots es el mecanismo para que unos widgets se comuniquen con otros.

- Algunos atributos como el texto de etiquetas y otros se modifican de modo similar al lenguaje HTML

- QT proporciona además otras funcionalidades:

- ◆ Librerías básicas para Entrada/Salida, Manejo de Red, XML

- ◆ Interfaces con bases de datos como Oracle, MySQL, PostgreSQL, ODBC.

- ◆ Plugins, librerías dinámicas (Imágenes, formatos,...).

Con Qt4 es posible desarrollar aplicaciones robustas y completas de forma ágil, unido a la eficiencia de la misma al estar escritas en C++.

Lenguaje para la implementación: C++

Para la implementación se analizaron varios lenguajes de programación, teniendo en cuenta que el software resultante debía ser multiplataforma, las alternativas estudiadas fueron C++, Java y C#. A pesar de que el C# es un lenguaje moderno y fácil de aprender, al estar ligado a la plataforma .NET, que solo se encuentra disponible para Microsoft Windows, fue completamente descartado. Existe una implementación libre del framework de Microsoft llamada Mono, sin embargo aún se encuentra en desarrollo.

Java por su parte es un lenguaje ampliamente utilizado en el desarrollo de aplicaciones multiplataformas, sin embargo su máquina virtual (JVM Java Virtual Machine) hace un uso excesivo de los recursos del ordenador, por lo que en computadoras no tan modernas –como son la mayoría de las existentes en Cuba- pudiera ocurrir que la aplicación resultante no se ejecute de forma óptima.

Se decide entonces utilizar C++ para llevar a cabo la implementación, debido a que es un lenguaje compilado que se acerca bastante a la programación a bajo nivel, permitiendo al desarrollador un control más avanzado sobre los recursos del sistema, y la creación de aplicaciones que consuman un mínimo de memoria. Unido a esto las librerías Qt4 están desarrolladas en este lenguaje, confirmando una vez más que es posible desarrollar aplicaciones en C++ que sean multiplataformas.

IDE para el desarrollo: Kdevelop

Entre los varios IDEs estudiados, se seleccionó KDevelop, por su mejor integración con la librería seleccionada (QT). Anjuta está dirigido a GTK y a desarrollos sobre el escritorio Gnome, Codeblock por su parte se orienta a wxWidgets.

KDevelop se integra completamente con las herramientas qmake y uic que brinda Qt4. Qmake es utilizado para la gestión de archivos del proyecto, el cual crea un archivo con extensión .pro con un registro de todos los archivos que componen el proyecto (archivos cabeceras .h, archivos .cpp, interfaces visuales .ui, recursos, etcétera), luego genera un MAKEFILE, que no es más que un conjunto de reglas para construir el proyecto, en este último archivo se definen librerías específicas para compilar, tipo de ejecutable, opciones extras del compilador, por otra parte la herramienta UIC se encarga de traducir a lenguaje C++ los archivos xml generados por el Qt Designer, permitiendo su uso en el proyecto. Todo este proceso es realizado por KDevelop de forma transparente al desarrollo. Otras de las ventajas que ofrece este IDE son: autocompletamiento y sugerencias de código, resaltado de sintaxis, explorador de clases del proyecto, administrador de recursos, edición de múltiples archivos, asistente para creación de clases, integración con el generador de documentación Doxygen.

Herramienta de Modelado: Visual Paradigm

Para el modelado de la solución se utilizará Visual Paradigm, herramienta profesional que soporta el ciclo de vida completo de desarrollo de un software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Entre sus principales características se encuentran:

- Soporte de UML versión 2.1
- Diagramas de Procesos de Negocio - Proceso, Decisión, Actor de negocio.
- Documento Modelado colaborativo con CVS y Subversion
- Interoperabilidad con modelos UML2
- Código a modelo.
- Código a diagrama.
- Ingeniería inversa Java, C++.
- Generación de código.
- Modelo a código, diagrama a código.
- Editor de Detalles de Casos de Uso.
- Diagramas de flujo de datos
- Generador de informes para generación de documentación
- Distribución automática de diagramas

Además es la herramienta utilizada en nuestra universidad en muchos proyectos de software libre, debido a que se cuenta con una licencia para su uso.

Subsistema de Gestión Visual

El subsistema de gestión visual de recursos multimedia para el Ambiente de Desarrollo Integrado (IDE) requirió una investigación profunda de posibles soluciones, sin embargo el escaso desarrollo de aplicaciones que cubrieran nuestras necesidades dentro de la naciente Industria del Software nos llevo a optar por el desarrollo de una solución propia. Algunas de las soluciones analizadas fueron:

F4L

El F4L fue un programa que se desarrolló como una propuesta para Linux del Macromedia Flash, solo que su divulgación no ha sido muy difundida. F4L avanzó mucho en la interfaz de usuario, logrando casi una copia del Macromedia Flash, sin embargo no tiene implementada casi funcionalidades, solo permitía el dibujo de algunas formas sencillas. Debido a problemas con su diseño este proyecto fue abandonado.

Qflash

QFlash fue una propuesta para realizar un clon de Macromedia Flash para Linux, esto se logró, principalmente, en la parte de la interfaz, que es en gran parte muy parecida a la de Macromedia Flash, pero aún carente de muchas de sus funcionalidades. Aunque QFlash no se terminó, logró desarrollar una interfaz muy parecida a la del Macromedia Flash. Integraba un panel de herramientas, línea de tiempo, inspector de componentes y un editor para Action Script (muy sencillo). Más tarde, QFlash se unió con el F4L para crear un programa conjunto: Uira.

UIRA

El proyecto UIRA combina los recursos y conocimiento de los proyectos F4L y QFlash, el mismo pretendió ser un completo IDE como alternativa al software propietario Macromedia Flash MX. UIRA es un acrónimo para UIRA Isn't a Recursive Acronym. Actualmente el proyecto ha sido cancelado debido a la ley DADVSI en Francia (RAFFARIN & AILLAGON, 2003). Se trabajó mucho en el diseño de clases, en el prototipo no funcional y en la organización del proyecto para evitar problemas como en el F4L; sin embargo no se llegó a implementar nada, quedándose en la fase de análisis.

KTOON

Ktoon es uno de los programas colombianos, realizado por la empresa Toonka Films, es Software Libre que funciona bajo plataforma Linux, su objetivo no es realizar un clon de Flash, sino un programa que ofreciera la funcionalidad de Flash en el SO Linux. Ktoon es un programa que permite crear animaciones en 2D del tipo cartoons y Anime, lo que lo hace una aplicación gráfica muy profesional; no solo sirve para hacer la labor de dibujo, sino también el render del mismo. El KToon constituye la alternativa más cercana al Flash pero aunque tiene muchas de sus opciones, cabe señalar que la interfaz del programa es distinta. Sin embargo no implementa un editor de código ActionScript ni tiene forma de añadirse a la película. Por lo que se queda como solo un software para el dibujo de animación.

PENCIL

Pencil 0.4.4b es un software completamente gratis para la animación y el dibujo, que te permite crear los tradicionales dibujos animados utilizando tanto mapas de bits como gráficos de vectores. Es un software específico para animaciones vectoriales orientadas a la web como Flash. Con Pencil 0.4.4b las animaciones pueden exportarse a archivos de tipo Flash y SWF. Sin embargo es un editor gráfico orientado específicamente al diseño y su sistema de animación cuadro por cuadro es totalmente manual.

SWFMILL

Es un traductor de XML a un swf y los swf en XML. La sintaxis es fácil y sirve para agregar objetos externos al swf con posterior utilización. Su uso más común es la generación de componentes de la librería conteniendo imágenes (PNG y JPEG), fuentes (TTF) u otras películas swf para su uso con los compiladores de actionscript MTASC (OSFLASH, 2006) o haXe. Además Swfmill puede ser usado para producir estructuras swf tanto complejas como simples. Fue construido basado en los procesadores XSTL/EXSTL (libxstl). Las entradas o salidas pueden transformarse en xstl que puede ser usado como XML o SWF binario. Usa comandos propios de XSTL para importar PNG, JPEG, TTF y SWF y para el mapeo de los números identificativos de los SWF. Construido con un lenguaje simple que soporta la creación de bibliotecas y la construcción de simples SWFs. Swfmill es Software Libre, bajo los términos de GNU General Public License.

SWFTTools

SWFTTools reúne un grupo de herramientas para crear y manipular ficheros swf (BÖHME, 2001). El mismo ha sido liberado bajo licencia GPL, y funciona en entornos Windows, Mac OS X, Linux y otros sistemas tipo Unix. La herramienta principal es SWFC, que recoge la descripción de la animación Flash en un lenguaje sencillo y genera el fichero de salida SWF. Es posible incluir scripts ActionScript en el fichero generado. Además incluye algunas herramientas para convertir el contenido de formatos JPEG, GIF, WAV y AVI en SWF, así como para extraer el contenido de ficheros SWF. La herramienta de preferencia en este grupo es el SWFC utilizada para crear SWF a partir de un script simple.

Después de un estudio de las mismas y tomadas las decisiones sobre el desarrollo de un producto propio se paso a materializar la idea. Los objetivos de este producto era desarrollar una aplicación que nos permitiera controlar todos los elementos visuales que se gestionan en una multimedia tanto externos como creados internamente. La solución obtenida se encuentra todavía en una fase de desarrollo por lo que carece de grandes funcionalidades. Hasta el momento podemos observar elementos que cumplen con los requisitos obtenidos en la fase de construcción del software tales como: interfaz grafica con un grupo de paneles comunes en este tipo de software como un panel de herramientas donde podemos encontrar las principales funcionalidades de sistema, un panel de propiedades donde se podrá editar y visualizar todos los elementos del producto que se desarrolla, una línea de tiempo para el control del flujo de tiempo de las multimedia, una biblioteca para gestionar todos los recursos externos de las multimedia y un escenario para la edición y creación de la multimedia. Todos estos elementos visuales no son más que el producto de una lista detallada de requisitos de este sistema que podemos resumir en:

- La gestión de proyectos.
- Modificación del directorio de trabajo.
- Gestión de propiedades del proyecto.
- La gestión de recursos en la biblioteca.
- La gestión de propiedades de un objeto.
- La gestión de la animación por fotogramas.
- Visualización del contenido en la escena.
- La gestión de herramientas básicas para el dibujo.
- Tratamiento de los colores de dibujado.
- Efecto zoom en la escena.
- Gestión de un historial de acciones.
- Exportación de un swf.

Finalmente una vista previa del producto hasta estos momentos:

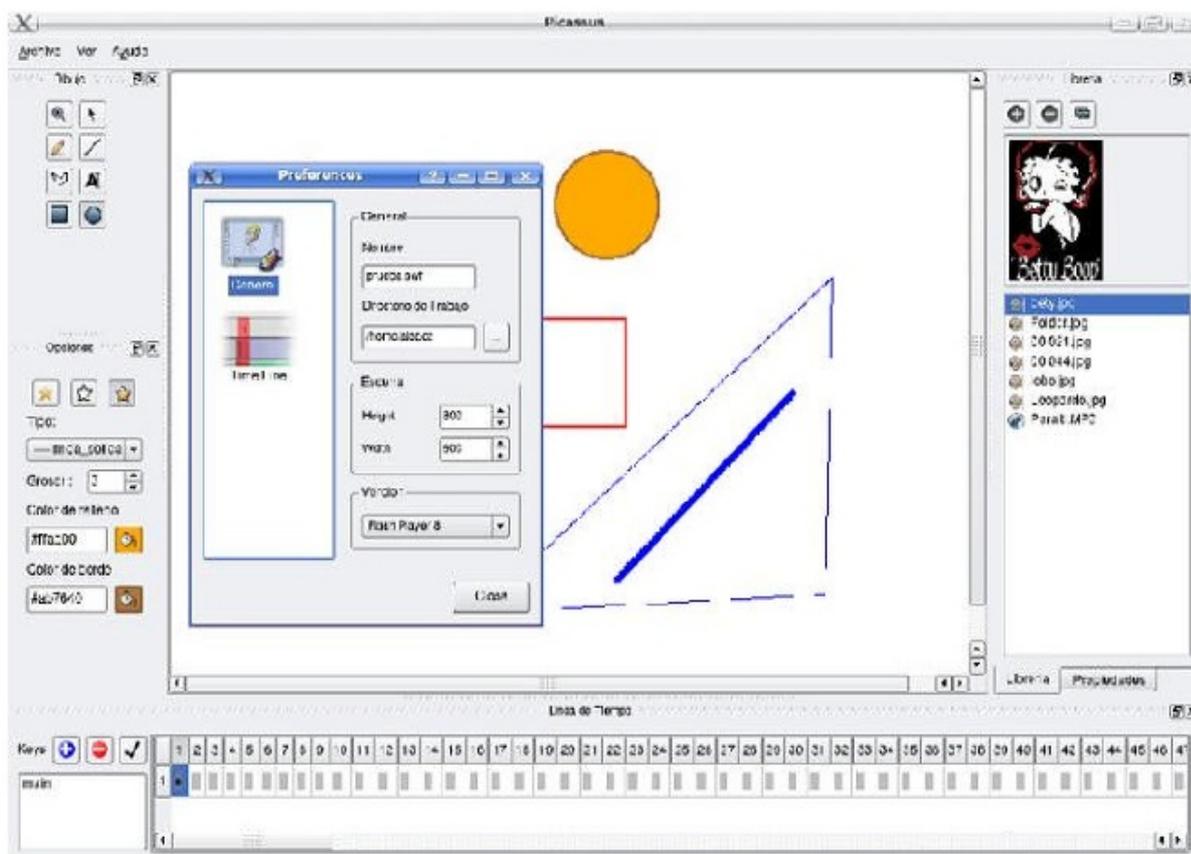


Figura 1. Interfaz principal en tiempo de edición de un proyecto

Subsistema de Gestión de Código

El Subsistema de gestión de código permite escribir código para multimedia (inicialmente ActionScript), para añadirle interactividad al producto que se esté implementando. Posee una arquitectura flexible basada en un sistema de plugins que permite extender sus funcionalidades sin modificar el código base de la aplicación, estas funcionalidades pueden ser: agregar nuevos compiladores al IDE (Ejemplo: haxe, flex 3.0, etc). Fue necesario hacer esta elección debido a que las herramientas de desarrollo de software educativo con tecnología multimedia evolucionan a diario, y aparecen con el tiempo nuevas tecnologías como el reciente Flex, las cuales no se podían obviar en futuros desarrollos.

Para su implementación se necesitó además realizar un profundo estudio de algunas de las herramientas existentes en el software libre y se analizaron características de IDEs similares pero solo disponibles para plataformas privativas como Microsoft Windows. Entre las herramientas estudiadas se encuentran:

Flash Develop: Es un completo editor de código para ActionScript, admitiendo hasta la versión 3 de este lenguaje. Es desarrollado usando el .NET Framework de Microsoft, que limita su uso solo a sistemas operativos de la familia Microsoft Windows. Utiliza como base para generar los archivos SWF el SWFMill y el MTASC (Mika Palmu 2008).

SEPY: SE|PY ActionScript Editor. Es un editor para ActionScript 2.0 orientado principalmente a plataformas Microsoft Windows, desarrollado en python. Incluye completamiento básico de código, resaltado de sintaxis, explorador de clases e integración con proyectos de Adobe Flash, entre otros, sin embargo desde hace dos años no cuenta con ninguna actualización, por lo que ha quedado obsoleto frente a las nuevas tecnologías y herramientas de desarrollo que surgen (Alessandro Crugnola 2006).

MTASC: Motion Tween Action Script Compiler. Es un compilador libre para ActionScript. Soporta hasta la versión 2.0.

Exige que el código que se escriba sea completamente orientado a objetos y es mucho más veloz que el compilador interno de Adobe Flash. No posee interfaz gráfica de usuario (GUI) que facilite su manejo, siendo una herramienta modo consola, dificultad que hace que algunos usuarios no lo utilicen por temor a complicarse. A pesar de que MTASC se encuentra en una versión estable, se ha dejado de desarrollar para dar paso a un nuevo compilador llamado haXe (Motion Tween 2007).

haXe: Este compilador ha sido desarrollado por los mismos programadores que hicieron el MTASC. Entre las características fundamentales que posee destacan:

- Soporte para ActionScript 1,2,3.
- Soporte para JavaScript.
- Posee además un lenguaje propio (haxe).
- Posee una máquina virtual llamada neko que permite agregar funcionalidades como escritura en archivos.
- No posee interfaz visual. (Motion Tween 2008).

Una vez realizado el análisis se decidió desarrollar un editor de código propio para el sistema en general, debido a que Flash Develop, el IDE más utilizado actualmente en desarrollos libres, se encuentra implementado sobre el Framework .NET de Microsoft, y ha sido imposible portarlo hacia plataformas libres, el SEPY que en determinado momento gozó de popularidad fue discontinuado, quedando entonces como alternativas MTASC y Haxe, dos potentes compiladores libres para multimedia, pero que no poseen interfaz visual que facilite la creación y gestión de proyectos de este tipo.

El Subsistema de Edición de Código cuenta con una interfaz visual sencilla e intuitiva, cumpliendo con las expectativas de cualquier sistema moderno, inspirado en algunos de los IDEs más populares como KDevelop y Flash Develop. Se utilizó además un componente llamado QScintilla, que permite brindar la funcionalidad de completamiento de código para varios lenguajes.

Sus características fundamentales:

- Sistema de plugins para extender sus funcionalidades. Asistente para la creación de proyectos.
- Explorador de Clases y Archivos del Proyecto.
- Panel Visualizador de errores ocurridos durante la compilación y ejecución del proyecto.
- Autocompletamiento de código basado en el lenguaje que se esté utilizando.
- Resaltado de sintaxis para las palabras claves del lenguaje.

A continuación imágenes del sistema resultante:

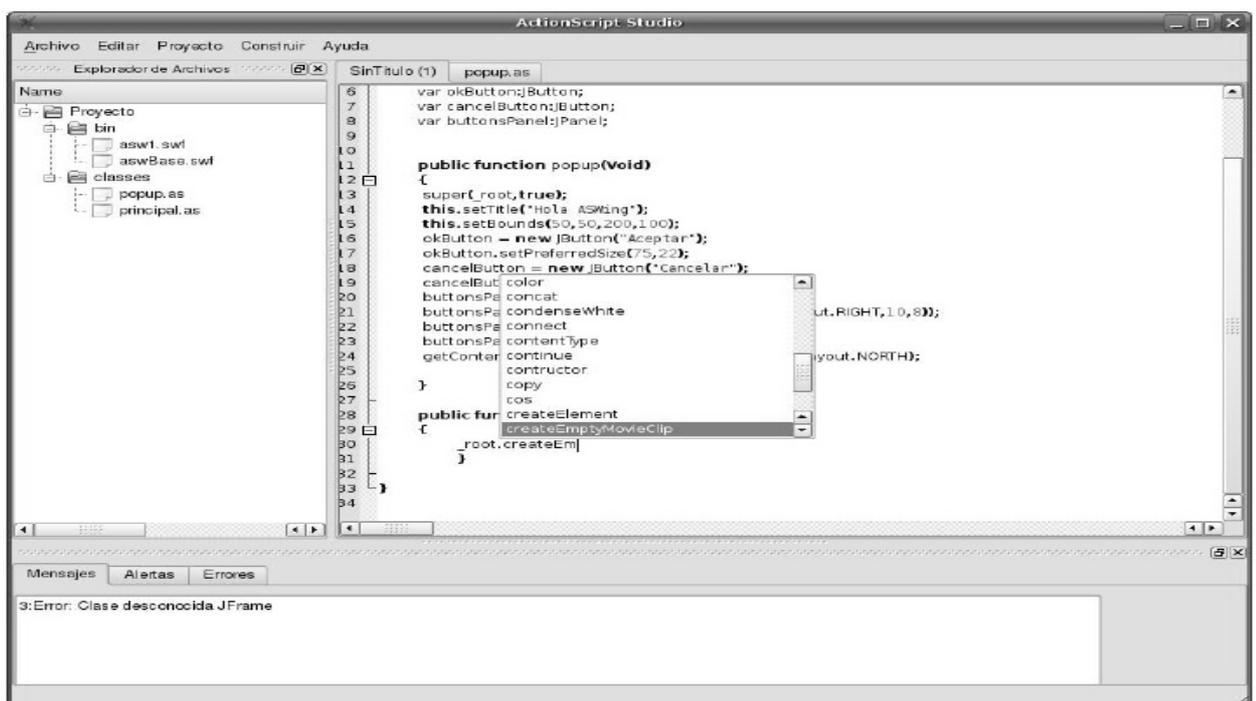


Figura 2. Interfaz principal en tiempo de edición de un proyecto

Subsistema Ejercicios y Clases Reutilizables

El presente subsistema tiene como objetivo principal proporcionar a los desarrolladores una serie de componentes, clases y ejercicios reutilizables para un rápido desarrollo de aplicaciones multimedia educativas e incluir estas clases en el IDE a desarrollar. En dicho subsistema se utilizó UML como lenguaje de modelado siguiendo la metodología RUP y apoyo en la herramienta case Visual Paradigm.

Como lenguaje de codificación fue escogido ActionScript 2.0 ya que es uno de los lenguajes más utilizados en este mundo de la multimedia, además posibilita en gran medida la programación orientada a objetos. Pero la principal razón por la que fue escogido es porque el IDE a desarrollar utiliza como compiladores principales MTASC y HAXE, estos compiladores soportan principalmente ActionScript y están orientados principalmente a la creación de películas flash siendo este formato uno de los más utilizados en la creación de multimedia.

Como librería gráfica se utilizó AsWing que es un framework de código abierto para ActionScript 2.0, permite a los programadores realizar aplicaciones utilizando una gran gama de clases y componentes con gran variedad de estilos y formas.

Como herramienta de desarrollo se escogió FlashDevelop por estar orientado a la programación en ActionScript, pero que no se limita a este lenguaje. También soporta JavaScript, HTML, XML y CSS. Es un entorno de desarrollo completo con todas sus funcionalidades: buscar y reemplazar, buscar en varios ficheros, debug, validación del código, etc. Además FlashDevelop soporta multitud de plugins que lo hacen aún más funcional.

Estas clases son totalmente dinámicas solo requieren de dos XML uno para la gestión de la configuración del ejercicio como son posición y tamaño de los elementos en pantalla, color, etc. También se requiere de un XML de contenido donde estará todo el contenido perteneciente al ejercicio.

Hasta el momento se encuentran implementadas varias clases que siguen las características anteriormente mencionadas, dichos ejercicios todavía se encuentran en una versión alfa la cual está en una fase de refinamiento tanto en diseño como en implementación.

Conclusiones

A pesar de que el producto de manera general todavía se encuentra en una fase de desarrollo y solo están implementadas las funcionalidades básicas, se concluye que el sistema brinda una solución factible a la problemática inicial del proceso de migración de la producción de software educativo y multimedia con herramientas libres. Su desarrollo y explotación significará una mejora considerable en las facilidades de producción nacional e internacional de software educativo en formato multimedia para la UCI. El sistema además significa un paso de avance en la búsqueda de la independencia tecnológica de la sociedad.

Referencias

Planchette, Jj. & Summerfield, M., 2006. C++ Gui Programming with Qt4, Prentice Hall.

Crugnola, A., 2006. SE|PY, the editor, the blog. SEPY, the editor, the blog. Disponible en: <http://www.sepy.it/>.

Palmu, M., 2008. FlashDevelop Open Source Flash. FlashDevelop Open Source Flash. Disponible en: <http://osflash.org/flashdevelop>

Motion Tween, 2008. haXe - ¡Bienvenido a haXe! haxe. Disponible en: <http://haxe.org/>

Motion Tween, 2007. Motion-Twin. Motion Tween. Disponible en: <http://www.mtasc.org/>

Martin Olivera, Y. P. & Saez Villavicencio, A. C. Análisis de un IDE para múltiples plataformas con tecnologías y herramientas libres para desarrollar software educativo en formato multimedia. Subsistema de gestión visual. Universidad de las Ciencias Informáticas, Ciudad de la Habana, 2008.

Zamora Sánchez, G. & Rodríguez Tamayo, Y. Análisis de un IDE para múltiples plataformas con tecnologías y herramientas libres para desarrollar software educativo en formato multimedia. Análisis, diseño e implementación de componentes, ejercicios y clases reutilizables. Universidad de las Ciencias Informáticas, Ciudad de la Habana, 2008.

Ferrer Obregón, R. & Jiménez Milian, J. P. Análisis de un IDE para múltiples plataformas con tecnologías y herramientas libres para desarrollar software educativo en formato multimedia. Subsistema de gestión de código. Universidad de las Ciencias Informáticas, Ciudad de la Habana, 2008.



"IV TALLER INTERNACIONAL DE SOFTWARE LIBRE Y ESTÁNDARES ABIERTOS DE SOFTWARE"

"IV International Workshop on Free Software and Open Standard Software"

Día / Day: 10 Feb/Feb 10 Sala / Room: 9
Sesión / Session: Mañana/Morning

Presidente / Chairman: Allan Pierra Fuentes
Secretario / Secretary: Ariel Reyes Antuan
Relator / Teller: Delly Lien González Hernández
TEMÁTICA: " Migración Software Libre. Experiencias comunitarias y de gobierno."

Hora / Time

Actividad / Activity

10AM



Inauguración / Opening

Presidente / Chairman: MSc. Héctor Rodríguez Figueredo.
Universidad de las Ciencias Informáticas / University of Informatics Science
Pte. Grupo Técnico de SWL de la República de Cuba / Pte. Technical Group of Free Software of Cuba.

10:10 - 11:30AM



Panel: "El Software Libre como alternativa gubernamental"

Moderador: Ing. Sergio García de la Puente
Universidad de las Ciencias Informáticas / University of Informatics Science
Gerente Grupo Investigación Migración UCI / CEO Free Software Migration UCI Group
Panelistas:
Ing. Ramón Paumier Samón
Universidad de las Ciencias Informáticas / University of Informatics Science
Gerente Grupo Investigación Migración UCI / CEO Free Software Migration UCI Group
Julio Vidal Larramendi
Universidad de la Habana / Havana University

11:30 - 1:00PM



Conferencia / Conference: "Experiencia de la Migración a Software Libre en la República Bolivariana de Venezuela"

Conferencista / Speaker: Carlos Eloy Figueira
Ministerio del Poder Popular para las Telecomunicaciones y la Informática / Ministry of People's Power for Telecommunications and Informatics
Pte. Centro Nacional de Tecnologías de la Información / President. National Center for Information Technology

Día / Day: 10 Feb/Feb 10 Sala / Room: 9
Sesión / Session: Tarde/Afternoon

Presidente / Chairman: Alfonso Alí Herrera
Secretario / Secretary: Eduardo Estevez Nuñez
Relator / Teller: María Caridad Valdés Rodríguez
TEMÁTICA: " Migración Software Libre. Experiencias comunitarias y de gobierno."

2:00 - 3:15PM



Conferencia / Conference: "La explotación de los derechos de autor en el software libre." / "The exploitation of copyright in the free software."

Conferencista / Speaker: Lic. Otto Batista Soler
Universidad de las Ciencias Informáticas / University of Informatics Science.
Abogado especialista en licencias de software libre / Lawyer specializing in free software licenses.



Hora / Time

Actividad / Activity

3:30 – 5:00PM

SWL008



Ponencias / Rapporteur

Nova, distribución cubana de GNU/Linux. Estado y perspectivas futuras.
Nova Cuban GNU/Linux distribution. State and future prospects.
Ing. Angel Goñi Oramas
Líder del proyecto Nova. Universidad de las Ciencias Informáticas. / Nova project leader. University of Informatics Science.

SWL019



Canaima una metadistribución de GNU-Linux para el Estado venezolano /
Canaima a GNU/Linux Distribution for the Venezuelan State.
Henry Rivero Proaño
Centro Nacional de Tecnologías de la Información

SWL014



Proyecto de Formación en SW Libre ETECSA Camagüey. / Project of
Open Source Training in ETECSA, Camagüey
Mefistoles Zamora Márquez
Empresa de Telecomunicaciones de Cuba

SWL029



Desarrollo y adopción de estándares abiertos en un marco de integración
regional. / Development & adoption of open standards in a regional's
integration framework.
Carlos González Rico
Centro Nacional de Tecnologías de la Información

SWL031



Distro Tinux: Una propuesta para los Jóvenes Club. / Distro Tinux: a proposal to
the Jóven Clubs.
Lic. Alberto Méndez Pérez
Hotel Brisas Covarrubias y Joven Club Pto Padre

SWL037



México

Clúster de Linux con enfoque de Alto Rendimiento para innovar la práctica
docente. / High performance Linux cluster for innovative learning.
Alicia Guadalupe Valdez Menchaca
Universidad Autónoma de Coahuila

SWL007



Resultados de la migración a software libre en la Facultad Regional UCI de
Ciego de Ávila. / Results of the free software migration strategy in the
Ciego de Avila's UCI Faculty
Disnel Rodríguez Rodríguez
Universidad de las Ciencias Informáticas

SWL0016



Reingeniería web para un portal de gobierno electrónico. / Reengineering Web
to Portal of Electronic Government
Palmariz Arrieta Jiménez
Ministerio de Energía y Petróleo

Día / Day: 10 Feb/Feb 10

Listado de ponencias para póster electrónico (se hará la programación centralmente).



SWL003

RADIGRAX: RECURSOS PARA APRENDER DISEÑO GRÁFICO Y REALIZAR AUDIOVISUALES
CON GNU/LINUX / RADIGRAX: Resources for Learning Graphic Design and Realization of
Audiovisuals with Gnu/Linux.
David Padrón Álvarez
Universidad de las Ciencias Informáticas



SWL006

Guía para la migración a software libre en empresas exportadoras e importadoras de equipos y piezas. / Guide for the migration to free software in business, exporting and importing equipment and parts.

Ing. Michel Hernández Rodríguez

Universidad de las Ciencias Informáticas



SWL009

ANÁLISIS SOCIO TECNOLÓGICO DE LA MIGRACIÓN HACIA SOFTWARE LIBRE EN LA FACULTAD 10 DE LA UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS. / TECHNOLOGY MEMBERSHIP ANALYSIS OF MIGRATION TO OPEN SOURCE SOFTWARE IN THE FACULTY 10 OF THE UNIVERSITY OF INFORMATICS SCIENCE.

Lic. Yenisleydi Cariaga Cristo

Universidad de las Ciencias Informáticas



SWL015

Un enfoque sistemático para la implantación de sistemas de información, desarrollados en software libre. / A SYSTEMATIC APPROACH FOR THE IMPLEMENTATION OF INFORMATION SYSTEMS, DEVELOPED IN FREE SOFTWARE

Palmariz Arrieta Jiménez

Ministerio de Energía y Petróleo



SWL018

Repositorio institucional de Información para la Universidad de la Habana. / INSTITUTIONAL REPOSITORY IN THE UNIVERSITY OF HAVANA

Edgar Bicet Alvarez

Universidad de la Habana



SWL020

Desarrollo económico sustentado por el Software Libre. Rompiendo un mito. / Economic development supported by Free Software. Breaking a myth.

Lic. Medardo Antonio Rodríguez Rodríguez

Centro Internacional de la Prensa



SWL021

Impacto del Software Libre en la enseñanza/aprendizaje de la programación. / Impact of Free Software in the teaching/learning of programming.

Lic. Manuel Vázquez Acosta

Universidad de las Ciencias Informáticas



SWL028

Las revistas de ciencias de la computación dentro del movimiento de acceso abierto. / THE JOURNALS OF COMPUTER SCIENCE INSIDE THE MOVEMENT OF OPEN ACCESS

Lic. Sergio Carbonell de la Fé

Universidad de las Ciencias Informáticas



SWL012

GeoTux, Soluciones Geoinformáticas Libres. / GEOTUX, FREE GEOINFORMATICS SOLUTIONS

Germán Alonso Carrillo Romero

